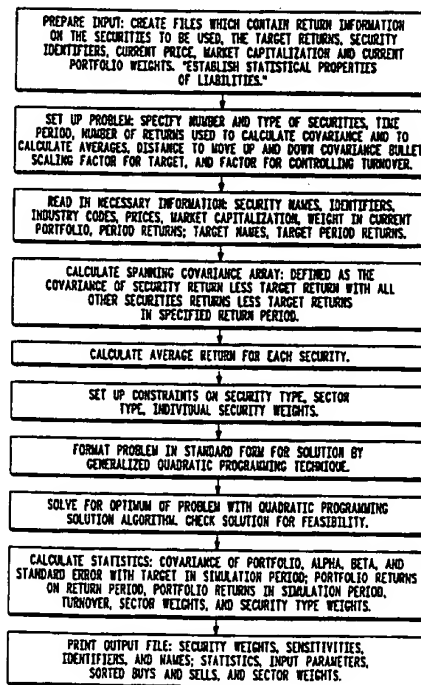




## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>5</sup> :  G06F 15/21		A1	(11) International Publication Number: WO 91/02326  (43) International Publication Date: 21 February 1991 (21.02.91)
(21) International Application Number: PCT/US90/04328 (22) International Filing Date: 2 August 1990 (02.08.90) (30) Priority data: 389,382                      2 August 1989 (02.08.89)                      US (71) Applicant: NATIONAL INVESTMENT SERVICE OF AMERICA [US/US]; 815 East Mason Street, Suite 1501, Milwaukee, WI 53201 (US). (72) Inventor: BAKER, Nardin, L. ; 2410 East Menlo Boulevard, Shorewood, WI 53211 (US). (74) Agents: RECHTIN, Michael, D. et al.; Reinhart, Boerner, Van Deuren, Norris & Rieselbach, 111 East Wisconsin Avenue, Suite 1800, Milwaukee, WI 53202 (US).		(81) Designated States: AT (European patent), AU, BE (European patent), CA, CH (European patent), DE (European patent)*, DK (European patent), ES (European patent), FR (European patent), GB (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent).  Published <i>With international search report.          Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>	

## (54) Title: RAPID METHOD OF ANALYSIS FOR CORRELATION OF ASSET RETURN TO FUTURE FINANCIAL LIABILITIES



## (57) Abstract

A method and system for correlating an expected asset return of a portfolio to changes in future financial liabilities and also to other financial indices. Management of asset portfolios often requires precise matching of liability streams, such as is the insurance industry and for pension funds. The method selects the weight percentages of assets by achieving optimum statistical correlation between asset returns and liability returns.

### DESIGNATIONS OF "DE"

Until further notice, any designation of "DE" in any international application whose international filing date is prior to October 3, 1990, shall have effect in the territory of the Federal Republic of Germany with the exception of the territory of the former German Democratic Republic.

#### FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MC	Monaco
AU	Australia	FI	Finland	MG	Madagascar
BB	Barbados	FR	France	ML	Mali
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GR	Greece	NL	Netherlands
BJ	Benin	HU	Hungary	NO	Norway
BR	Brazil	IT	Italy	PL	Poland
CA	Canada	JP	Japan	RO	Romania
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	LI	Liechtenstein	SN	Senegal
CM	Cameroon	LK	Sri Lanka	SU	Soviet Union
DE	Germany	LU	Luxembourg	TD	Chad
DK	Denmark			TG	Togo
				US	United States of America

RAPID METHOD OF ANALYSIS FOR CORRELATION OF  
ASSET RETURN TO FUTURE FINANCIAL LIABILITIES

The present invention is related generally to a method and system for selecting a portfolio of assets for achieving optimum correlation of asset return to a selected standard financial index. More particularly, the invention is related to a highly efficient, rapid method and system for choosing an asset portfolio having the optimum correlation of the asset return to a time dependent financial index, such as a financial liability, at each of a number of selectable asset return levels.

Management of portfolios of assets has historically emphasized maximizing the return on assets with the objective of at least outperforming the market. However, in some financial industries the objective, or the figure of merit, is also related to meeting future liabilities rather than just achieving high return on assets. Frequently, an institution will have a future intended use of the assets which requires their availability at some future time. If assets are performing differently than liability requirements, substantial financial hardship can result. For example, insurance companies and corporate pension plans have well defined future financial liabilities which must be met. Consequently, although return on assets is one important objective, meeting future liabilities is also important and can be even more

2

important in many instances. In fact, many pension plan managers are now required to meet the standards set forth in FASB Statement 87 (Financial Accounting Standards Board) on pension fund accounting. Under the FASB Statement a market interest rate return on pension funds is the standard index and is to be based on A-rated ten year corporate bonds. Under this FASB Statement any deficit in corporate pension funds are now reflected on the balance sheet. Any such deficit would therefore have substantial adverse effect on the apparent net worth of the subject corporation. Consequently, this FASB Statement standard strongly encourages maintenance of a surplus for a pension fund. As an example of the importance of matching the liability requirements under the FASB Statement, consider the percentage change possible for pension plan liabilities, as measured by the accumulated benefit obligation (ABO). If, for example, interest rates increase by 1% in one year over the present rates, the present value of the ABO would decline by 10% if the ABO has a duration of 10 years. Likewise, if interest rates were to drop by 1% in one year, the present value of the ABO would increase by 10%. The potential for such dramatic fluctuations in liabilities clearly deserves careful attention by parties obligated to meet future liability streams.

In order to timely meet future financial liabilities and maintain a proper surplus fund for a pension plan, a number



of methodologies have arisen including "immunization", "cash matching"; and some preliminary efforts have even been directed to utilizing stock funds.

The "immunization" method of meeting future financial liabilities uses bonds having substantially the same duration as the liability stream. Duration is a measure of volatility expressed in years, which is similar to, but more precise than, average life. The duration is calculated as the weighted average amount of time to the receipt of the payout. There are however significant drawbacks to "immunization", with one primary disadvantage being the relatively low excess return on assets generally achieved by the method. Additional limitations are imposed by the two major assumptions made by the strategy: The yield curve (a plot of yield to maturity on bonds versus their time to maturity) will only make parallel shifts. Consequently, regardless of maturity, when market conditions change, all bonds allegedly move exactly the same amount in yield. This clearly is not the case since there have been substantial inconsistencies in the past for the difference in rates for short-term bonds and long-term bonds. Secondly, all cash flows in excess of required annual payments can allegedly be reinvested at the yield to maturity of the portfolio. This presumption is also clearly not true since sharply declining or rising interest rate environments will make it extremely difficult to carry out reinvestment.

4

Furthermore, this strategy does require more ongoing management of the portfolio in order to sell or buy more securities to match the actuarial schedule and maintain a proper asset/liability match.

The "cash matching" method utilizes a bond portfolio having numerous component bonds with various maturity dates and payout rates to precisely match the liability requirements of the pension plan. Such an approach has the same primary disadvantage as the "immunization" method and further requires additional effort to assemble the portfolio. Frequently, the "cash matching" method demands payment of a premium to achieve the correct mix of bond rates and maturity. Both of the first strategies ("immunization" and "cash matching") must invest in fixed income securities to provide the assurance of receiving the necessary cash flows. In fact, they must primarily invest in U.S. treasury obligations since investments in corporate or mortgage securities increase the chance for default or for call risks which can have the effect of changing the projected cash flow.

Pension plan liabilities or other future liabilities, such as are present in the insurance industry, are long term in nature. Therefore, a future liability stream can greatly benefit from the compounding effect of investment in higher returning assets, such as common stocks. However, attempts to characterize stocks in terms of a time duration parameter or

5

otherwise have not been successful. In the last few years many unsuccessful attempts have been made to develop a system whereby a portfolio of equities is linearly optimized relative to a liability stream. There have been attempts to parallel the "cash matching" techniques with the use of stocks, instead of bonds. This approach has involved matching the expected dividend flow of the portfolio to the liability stream. Unfortunately, stock dividend yields are unpredictable, particularly beyond 3 years in the future. Another major effort in equities has been directed to an "immunization" type treatment. In this effort an attempt was made to calculate the duration of stocks on an individual basis, as well as on a portfolio basis; but these attempts also have been unsuccessful, primarily due to the long term unpredictability of stock dividends.

In a related patent application, incorporated by reference herein and having serial number 281,560 and filed December 8, 1988, an improved method and system were set forth directed to correlating return on assets to a financial objective over time. In performing the analysis to determine the optimum assets of a portfolio to track the financial objective, the machine time and efficiency of the evaluation process can limit the number of assets considered in constructing the portfolio. Such limitations on the number of assets which are considered for inclusion in the optimum

6  
portfolio can also limit the performance of the selected portfolio.

It is therefore an object of the invention to provide an improved method and system for determining the optimum portfolio of assets for tracking a financial index.

It is another object of the invention to provide a new method and system of efficiently selecting the optimum portfolio of assets for tracking a financial index.

It is an additional object of the invention to provide an improved method and system of rapidly analyzing a large number of potential assets to select the optimum portfolio of assets to track a financial index.

It is a further object to provide a new method and system of enlarging the number of potential assets under consideration for inclusion in a portfolio of assets, while reducing the time required to select the portfolio of assets which best track the behavior of a financial index.

It is another object of the invention to provide a rapid, more efficient method and system of selecting the weighted values for assets selected from a universe of possible assets for a portfolio designed to track a financial index.

It is an additional object of the invention to provide an improved method and system for reinvesting cash flow from a portfolio starting with that current portfolio of assets.

7

Further objects and advantages of the present invention, together with the organization and manner of operation thereof, will become apparent from the following description when taken in conjunction with the accompanying drawings described hereinbelow.

#### BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1A is a functional flow chart illustrating operation of one method of portfolio construction and FIG. 1B is a flow chart illustrating operation of a particular method of the invention;

FIGURE 2A illustrates simulation results of annual total liability returns of a pension plan (dashed) and annual total asset returns (solid line) for a preferred form of the invention and FIG. 2B shows annual total liability returns of the simulated pension plan (dashed) and annual total asset returns (solid line) for the Standard & Poors 500;

FIGURE 3A is a bar graph of simulation results for funded status returns over time for a preferred method of the invention and FIGURE 3B is a bar graph of funded status returns over time for the Standard & Poors 500;

FIGURE 4 a comparative plot of cumulative funding status for simulation results over time for a pension plan liabilities (dashed), a portfolio derived by a preferred method of the invention (solid line) and the Standard & Poors 500 index (dashed and dotted);

8

FIGURE 5A shows the correlation between asset return and liability return for a preferred method of the invention and FIG. 5B illustrates correlation between asset return and liability return for the Standard & Poors 500 index; and

FIGURE 6 illustrates the boundary line of minimum risk for various future asset return levels.

#### DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Broadly stated, a method and system are described for selecting a portfolio of assets and correlating a future asset return of the portfolio to a financial index, such as, a liability index, an inflation index, or any other accepted index and mixtures thereof. Specific examples of indices are liability indices, such as, individual corporate pension plan liabilities and insurance company liabilities. The consumer price index and wage growth index are examples of an inflation index, and other indices can include accepted stock price indices and futures markets indices. The method includes selecting asset portfolios which optimally correlate portfolio returns to the future desired payouts or payments needed over time to fulfill the desired financial objective. In the general case the user selects a standard index to which optimum correlation is desired for the selected portfolio having a future asset return over time. The process of selecting the

9

standard index can involve obtaining input (such as actuarial) in terms of the characteristics of future cash payments discounted to present value based on a range of discount rate and wage (inflation) values. This information can be used to construct a functional behavior for the present value of the liability. A decision is then made, such as by a company pension fund manager, that certain discount rates and inflation assumptions should be made. On this basis the current liabilities are projected back in time using these assumptions and a plurality of assets are examined to determine their sensitivity to the past behavior of the liability returns. In the most general sense if one can determine an index to which a portfolio of assets has a strong correlation, this sensitivity can be used to select a set of assets which will match the behavior of the index as it changes over time. As a particular example an actuary can provide specific ranges of present value liability for a range of discount rates and inflation rates. The change over time of the liability from month to month over a twenty four month period can yield a liability return. The analysis to be described in more detail hereinafter determines which selected ones of a plurality of assets track the liability returns with best correlation. The resulting weighted set of assets form the portfolio to follow the future liability returns. An analysis using the selected standard index can be performed on a plurality of assets, such as, for

/C/

example, at least one of the following categories of assets: stock securities, real estate investments, futures contracts, options, commodities, currencies and precious metals. The analysis allows the identification of the combination of weight percentages of selected ones of the plurality of assets yielding the optimum correlation of the future asset return to the standard index. Optimum correlation is thus achieved by calculating a minimum standard deviation or a variance for the difference between the return of the portfolio of assets and the selected standard index return. This method and system are particularly applicable for, but not limited to, the insurance industry and management of pension fund liabilities.

FIGS. 1A and 1B illustrate in functional flow charts the procedures followed in carrying out two forms of the invention. In the first method shown in FIG. 1A (and described previously in pending application having serial number 281,560) the correlation of the expected asset return of a portfolio to a standard index one is initiated by input of various basic information. This information includes, for example, establishing the fundamental statistical characteristics of liability returns, and future payment schedules for matching a desired index, such as the future stream of financial liabilities of a pension plan. As described hereinbefore, the future payment schedule for a pension plan can be determined by using actuarial data. These future liabilities can be



//

characterized in terms of an accumulated benefit obligation (ABO), that is, the price you would have to pay if the liabilities were sold at a selected time. The total outlay required to pay retirement wages for the pension plan are discounted back to the present value at the market rate interest (currently 10%). Other related characterizations can be used, such as a projected benefit obligation (PBO), by accounting for inflation in the growth of wages at retirement. This amount is converted to a percentage and an expected salary at retirement, discounted to present value. Therefore, although the ABO is affected primarily by interest rates, additional standard measures, such as the PBO, account for inflation. Therefore, the method is also generally effective for calculating the convolution of complex effects with one another. The method only requires optimizing correlation of the time behavioral performance of future asset return relative to the particular standard index, which includes any conceivable selected characteristic which assets are found to be sensitive to.

In the manner illustrated in step 1 of FIG. 1A, various input files are therefore created to begin the analysis. These input files can include, for example, asset return information for the universe or plurality of assets to be sampled in the analysis. Also established as data files are the data representative of the standard asset return over time,

12

such as target returns for a future liability stream of a pension plan or an insurance company. The future liability stream can depend on interest rates and/or inflation rates and other variables which can affect the liability stream. For example, as described hereinbefore, a surface can be generated which describes the behavior of liability return as a function of both interest rates and inflation rates. Other information in the data files can be identification information for the plurality of assets, current price and market capitalization of the assets, as well as the characteristic weight percentages of assets in a previously selected portfolio. Weight percentages, for example, from a prior period would be used in the most preferred embodiment.

As illustrated in the second box of FIG. 1A, the number and type of the plurality of assets should be specified, the time period for matching the standard, the number of returns used to calculate a statistical correlation and the number of asset returns used to calculate averages for the plurality of securities to be analyzed to select the optimum set of weighted assets.

The method in FIG. 1A then initializes information preparatory to analyzing the plurality of assets, such as, establishing names of securities, associated identifier information, industry codes, prices of securities, market capitalization, weight and percent of the previously calculated

13

prior portfolio, the period for the asset return, the name of the target or standard for measuring a standard asset return and the time period for the standard returns.

In the next box in FIG. 1A the correlation between the future asset return and the standard index is optimized by first generating a covariance array. While other nonlinear statistical analyses are possible, this method being described is a preferred method of carrying out the analysis. For example, another useful statistical method of analysis is correlation parameterization which is embodied in the computer software program Appendix III. As shown in the next step in FIG. 1A, the average return of each security is calculated followed by imposing certain constraints on the calculation, such as setting a range of weight percentages to be tried. The calculation is then implemented to a solution by a standard computer program quadratic technique (see Appendix I). This step is then followed by determination of various statistical parameters, such as X and B, standard error, portfolio returns over various time periods and for selected weights. The analysis is then completed by printing output (see attachment to Appendix I) such as asset weights, sensitivity factors for selected assets of the portfolio, statistical parameters, sorted buy and sell orders and sector weights.

A simple example of utilizing the preferred statistical method is illustrated for a portfolio containing

14

three stocks (designated 1, 2, and 3). In order to find the optimum weight percent for each of the three stocks in the portfolio, the minimum standard deviation (square root of variance) is calculated for the differences between the assets of the portfolio and the future liabilities as represented by the standard asset return over time. The risk is therefore defined as the standard deviation of differences:

$$\text{Risk} = \left[ \frac{\sum_{i=1}^n (R_{pi} - R_{Ti})^2}{(n-1)} \right]^{1/2} = \sigma[R_p - R_t] = \text{variance}^{1/2}$$

where:  $R_{pi}$  = total return on the portfolio during period  $i$ ;

$R_{Ti}$  = total return on target or standard in period  $i$ ,

$\sigma$  = standard deviation;

$R_p$  = average return on portfolio,  $i = 1, n$ ; and

$R_T$  = average return on target or standard portfolio of assets,

$i = 1, \dots, n, \dots$

The portfolio return equals percentage weight for each stock times the return on that stock:

$$R_{pi} = \sum_{j=1}^3 x_j R_{sj, i}$$

$x_j$  = the weight in the portfolio of the stock  $j$

$R_{sj, i}$  = the return on stock  $j$  in periods  $i$

Now in the definition of risk, as set forth above, we can substitute the following:

$$R_{pi} = \sum_{j=1}^3 x_j R_{sj, i}$$

15

Making this substitution, a determination of risk in the manner set forth above results in the calculation of the covariance of the stock with each of the other stocks in the portfolio after subtracting the return of the target, or standard index, from the future asset return of each stock.

The covariance of stock 1 with stock 2 is therefore:

$$(R_1, R_2) = \sum_{i=1}^n (R_{s1i} - R_{Ti}) (R_{s2i} - R_{Ti}) - n (\bar{R}_1 - \bar{R}_T) (\bar{R}_2 - \bar{R}_T)$$

We calculate all the spanning covariances and put them in a matrix form:

$$\begin{bmatrix} \text{Cov} [(R_1 - R_T), (R_1 - R_T)] & \text{Cov} [(R_1 - R_T), (R_2 - R_T)] \\ \text{Cov} [(R_2 - R_T), (R_1 - R_T)] & \text{Cov} [(R_2 - R_T), (R_2 - R_T)] \end{bmatrix}$$

In order to calculate the risk, we add up all the Cov terms times the weights in each stock:

$$\text{Risk} = \sigma_{(R_p - R_T)} = \left\{ \left[ \sum_{i=1}^3 \sum_{j=1}^3 X_i X_j \text{Cov} [(R_i - R_T), (R_j - R_T)] \right] \right\}^{1/2}$$

In order to minimize this "risk" function, we determine the combination of weight percentages for stocks 1 thru 3 which produces the smallest statistical risk. The above described risk can readily be calculated by various means, such

16

as, by a computer program (which is included in Appendix I). The output (see attachment to Appendix I) of the calculation includes the weight percent of each stock and the associated overall risk level. This calculation can be repeated for a range of expected asset return levels and results in generating a nonlinear type "bullet" shape defining the limits of minimum risk over a range of asset return levels for associated standard deviations of funding level (see FIG. 6). The method uses historical returns for the plurality of stocks analyzed in order to calculate the resulting covariance between the standard liability returns and the future returns of the potential portfolio of assets. Appendix II illustrates an example of a computer program for calculating typical liability return data. The method of analysis results in choosing a selected set of assets for the portfolio with a strong inclination of the selected set of assets to respond in a manner such as the standard asset returns over time, which alone can be valuable output. As mentioned hereinbefore, in other embodiments, the nonlinear analysis of a plurality of assets can involve other methods, such as, index correlation parametrization for matching the performance of a target index return (see Appendix III).

In one embodiment of the invention illustrated in FIG. 1B, the method is a simplification of the more formal procedure of FIG. 1A. The method of FIG. 1B accomplishes,

17

however, the same result as in FIG. 1A but with much greater efficiency and speed.

The method of FIG. 1B is broken into eleven steps, and the first three steps are substantially the same as the method of FIG. 1A. The eleven steps and details of each each step are described below:

I. Read in Returns of Selected Securities. Large numbers (thousands of securities) can be utilized in this procedure. An input file contains the returns for each security in the prior periods, such as the previous 24 months. Various data is included, such as, security name, industry groups code, market capitalization, trading volume, recent prices, specific identifiers and estimated bid/ask price spreads. The effect of transaction costs associated with the spread in the bid/ask price can be included in the performance analysis.

II. Read in Returns of Target Index to Track. If a portfolio is to be constructed for tracking a specific financial target index, the returns to that target for the relevant period are read into memory arrays. Additional identifying information is also read in from the target data file.

III. Set-up Constraints on Upper and Lower Bounds in Terms of Percent of Portfolio for each Security and each Sector or Industry. A maximum and minimum percentage weight of

18

the portfolio for each security can be specified to constrain the portfolio. This can be used to insure portfolio diversification and to control costs associated with trading. In addition, sectors of the universe, for example, utility stock, can be constrained by maximum and minimum boundaries. If one has a single target, one can "short" a stock and take a negative minimum position.

IV. Calculate the Covariance of Each Security with the Target Index. If a target is used the covariance is calculated for each security and stored in an array. If no target index is used, a zero value for each security is stored in the array.

V. Create an Initial Portfolio by Selecting Highest Covariance Securities and Weighting Them at their Upper Limits as Defined by Selected Constraints. This step creates an initial feasible solution to the problem by filling the vector of portfolio weights according to the constraints and in order of highest covariance.

VI. Calculate the Objective Function Value at Initial Portfolio Weightings. The objective function can be defined in a number of ways. The computer program allows monthly or moving quarterly returns to be used for optimization. Transaction costs can be considered and their importance magnified or reduced relative to other objectives. With minor changes, other such goals can be incorporated into



19

the objective function. The key is that once the objective function is specified, partial derivatives can be used to guide the search for an optimal portfolio. Any example of a preference that can be created is an increased weighting for a stock with a likely dividend versus one with no dividend.

VII. Calculate Partial Derivatives for Each Security at Initial Portfolio Weightings. A partial derivative is calculated for each variable (in this case each security), and a direction can be determined in which to move the individual security weights in order to obtain an improved portfolio solution. The partial derivatives are also used to determine if the weights are optimal. In the prior art, the solution techniques require storage of a full covariance matrix array. This storage requirement has limited the practical number of securities which could be considered at one time in the past methodologies. Indeed, the storage requirements, and to a large degree the processing time, varies as the square of the number of securities in the portfolio under consideration. See the example discussed hereinafter in which the previous methodology is compared to the invention.

In addition, the methods of solution for these problems in the prior art were slow and cumbersome and subject to failure when the full covariance array was sensitive or a nonunique set of solutions were achievable. This current method requires much less storage, uses a rapid solution

20

technique and allows control of the tolerance used for optimality. Consequently, the improved methods will select a portfolio when several combinations are equally desirable.

Not only is the amount of necessary memory reduced and the computer calculational time greatly reduced, there is substantial flexibility in defining the objective, assurance of the solution is enhanced and simultaneous considerations of large number of securities allows substantial improvement in optimizing the expected return of the portfolio compared to the target index.

VIII. Change of Portfolio Weightings by Moving in Direction of Improvement is Indicated by Partial Derivatives.

Adjustment of the portfolio weights is achieved by a search technique which moves along the constraints and changes in a proper direction of improvement of the objective function. The objective function is calculated at the new weights and a test of improvement is made.

IX. Test for Convergence to an Optimal Solution Defined by the Kuhn-Tucker Conditions. If the objective function is not improving or if the step size used to adjust the portfolio weights becomes extremely small, the search is terminated. This solution is normally a Kuhn-Tucker point (conventional method of establishing optimality conditions) or extremely close thereto within an acceptable epsilon to such a point.

21

X. Recycling Conditions. If the termination conditions are not satisfied then one re-calculates the objective function value, re-calculates the partial derivatives, makes changes in the portfolio weightings to achieve an improved solution and test for convergence.

The process is by nature iterative and continues cycling until a solution is reached in which diminishing returns are achieved by further cycling.

XI. Output of Information. Relevant portfolio information is output with security weightings, objective function values, purchases and sales necessary to achieve the optimum portfolio and industry weightings.

The detailed output is written into a computer file which then can be examined for relevant information. Order to buy and sell securities can be developed from the information in the output file.

Included in Appendix IV is an exemplary computer software (source output) program illustrating critical steps of the method of FIG. 1B. Table XI shows exemplary results for a program simulation wherein the target index is the Standard and Poors 500 stock index. Appendix VI illustrates significant distinctions from the optimizer methodology used in the copending patent application having serial number 281,560.

#### Quantifying Magnitude of Reduced Computation Time

A test was performed on an IBM compatible PC to

22

compare the solution speeds of two portfolio optimization systems. In a prior system, the problem of handling large numbers of securities in a portfolio selection process increased in proportion to the number of securities squared. Thus, a problem involving one hundred securities would take approximately one hundred times as many calculations to solve as a problem with ten securities.

In the current system the solution difficulty increases by a factor of less than one times the number of securities. In addition, the computing memory required to solve the problem is proportional to the number of securities rather than, as in the prior system, that number squared.

#### Solution Time Comparison

Hardware:     IMB compatible PC, 386-20Mhz CPU, 80387  
                  co-processor

Problem Size: 100 securities

3% portfolio weight upper bound on each security

0% portfolio weight lower bound on each security

Time to solution:

Prior system - 9 minutes 34 seconds

Current system - 52 seconds

Overall speed-up: 11.0 x prior system

23

This speed increase is actually of a larger magnitude since approximately 40% of the 52 seconds to solution with the current system is spent reading information from external files and writing information to other files. The solution time speed-up, allowing for reading and writing files, is approximately 17.3 x prior system. This speed-up ratio increases in proportion to the number of securities considered squared.

The speed increase and the reduced computer memory requirements by the current system allow large problems to be solved in a short time, requires limited computer memory, and uses computer hardware which is relatively inexpensive.

Circumstances arise regularly in the investment field which rapidly change the prospects for securities. The impact of these sudden changes must be incorporated into the security valuation system so that rational alternations in the investment portfolios may be made.

Examples of sudden changes include: a company is presented with a buy-out offer by another firm; a disaster occurs, such as an oil spill, which may impact a firm's stock price; monetary or fiscal policy changes are implemented by the government. It is important for an investment system to be flexible and fast enough to evaluate the impact these changes may have on a security portfolio.

24

The current system allows for estimates of partial monthly returns to be calculated on any day of the month, for these returns to be used in the optimization process, and for the results of the analysis to be completed within a few minutes.

One advantage of the current system is that analysis of the current investment opportunities can be completed rapidly and recommendations for buying and selling securities can be quickly generated. This allows investment decisions to be made and implemented quickly with confidence.

Further illustrations of the invention are exemplified by various historical simulations shown in FIGS. 2-4 and Tables I-III which are taken over the time period of 1975 to 1987. As listed in Table I and in FIGS. 2 and 3, the liability stream for a selected pension plan can undergo substantial variation with time. A portfolio of assets has been analyzed in accordance with the preferred statistical method described hereinbefore, and details of the selected portfolio are set forth in Tables IV-X. Over the 1975-1987 time period, the resulting portfolio of assets shows substantially better correlation to the liability stream as compared to the Standard & Poors 500 return. Moreover, as seen in Table II and FIG. 4, the overall cumulative return for the portfolio of assets selected by the preferred method is far better than the Standard & Poors 500. The greatly enhanced stability and good

25

TABLE I  
Funding Return Analysis

Year	Total Spanning Return (A)	Total S&P 500 Return (B)	Liability Return (C)	Spanning Funded Status Return (1+A)/(1+C)	S&P 500 Funded Status Return (1+B)/(1+C)
1975	28.85%	37.36%	7.43%	19.94%	27.86%
1976	34.96	23.94	15.53	16.83	7.28
1977	-4.16	-7.29	1.16	-5.25	-8.35
1978	4.76	6.42	-0.27	5.05	6.71
1979	21.84	18.36	-0.56	22.52	19.03
1980	13.68	32.34	-2.64	16.76	35.92
1981	16.01	-4.95	2.23	13.49	-7.02
1982	32.60	21.49	39.64	-5.04	-13.00
1983	22.15	22.43	0.46	21.59	21.87
1984	6.86	6.18	14.97	-7.05	-7.65
1985	40.95	31.59	31.33	7.33	0.20
1986	30.04	18.64	26.07	3.15	-5.89
1987	0.87	5.28	-5.05	6.24	10.88

26

TABLE II

## Spanning Technology

## Historic Simulation

## Cumulative Surplus Analysis

Year	Spanning Portfolio		S&P 500 Index	
	Cumulative Dollar Value	Cumulative Funded Position	Cumulative Dollar Value	Cumulative Funded Position
1975	\$100.00	100.00%	\$100.00	100.00%
1976	118.78	119.65	127.12	128.85
1977	149.36	140.49	147.26	138.52
1978	133.59	133.82	126.88	127.10
1979	130.14	141.83	125.03	136.25
1980	148.14	177.42	137.62	164.83
1981	157.98	216.50	170.79	234.06
1982	173.32	263.84	152.84	232.66
1983	218.19	267.96	173.99	213.68
1984	256.52	348.87	202.88	275.93
1985	263.48	350.40	205.10	272.76
1986	359.74	403.26	258.85	290.17
1987	457.65	438.20	297.13	284.50
	453.01	496.15	304.26	333.22

Note: The above cumulative values reflect monthly payments to beneficiaries.



TABLE III  
Spanning Technology  
Historic Simulation

Year	Spanning Portfolio Return	Beta	Alpha	*1975 to 1987 Results:	Spanning Portfolio	S&P : Inde
1975	28.85%	0.778	0.04%	Alpha - risk adjusted	4.33%	0.0%
1976	34.96	0.818	11.67	Beta - risk adjusted	0.77	1.00
1977	-4.16	0.940	2.59	Average Return	19.2%	16.3%
1978	4.76	0.859	-1.76	Correlation to Liability	69.6%	33.3%
1979	21.84	0.956	3.31	Return Volatility	13.6%	13.6%
1980	13.68	0.569	-8.15	Funding Return Volatility	10.1%	15.0%
1981	16.01	0.876	17.80			
1982	32.60	0.829	10.39	<u>Beginning Portfolio (1/75)</u>		
1983	22.15	1.168	-2.03	Dividend Yield	5.53%	3.41%
1984	6.86	0.832	0.11	P/E	7.2	7.7
1985	40.95	1.052	6.09			
1986	30.04	0.952	10.09	<u>Ending Portfolio (1/87)</u>		
1987	0.87	0.887	-4.64	Dividend Yield	5.23%	4.0%
				P/E	16.8	16.0

Duration of Liabilities at  
a 9% Interest Rate 9.61 yrs

\*Note: results are based on annual returns

Electric, Gas, and Sanitation Utilities  
and Banking Restricted to 10% of Portfolio.

28

TABLE IV

# SPANNING PORTFOLIO COMPOSITION 1987 Portfolio

<u>Industry</u>	<u>% of Portfolio</u>
Food Kindred Products	12.10
Textile Mill Products	2.57
App. & Oth. Fin. Prod.	.80
Furniture and Fixtures	2.39
Printing Publishing and A.P.	2.73
Chemical and Allied Prod.	8.85
Primary Metal Industries	.81
Fab. Metal Industries	1.34
Machinery Except Electrical	8.42
Ele. and Ele. Mach.	2.50
Meas. Anal. & Cont. Inst. Etc.	6.76
Transportation By Air	5.16
Communication	2.89
Electric Gas And Sanit. Serv.	10.00
Wholesale Trade--Durable Goods	.46
Wholesale Trade--Nondur. Goods	1.56
General Merch. Stores	1.88
Food Stores	3.00
Eating And Drinking Places	2.40
Banking	7.17
Cred. Agen. Oth. Than Banks	3.00
Insurance Carriers	2.36
Hold. and Other Inv. Comp.	2.90
Hotels Room. Houses Camp AOLP	.17
Health Services	1.76
Miscellaneous Services	3.00
Nonclassifiable Establishments	3.00

29

TABLE V

SIMULATION RESULTS  
1975  
HISTORIC SIMULATION:

LIABILITY STREAM USED LONG TERM STUDY ABO

NUMBER	WEIGHT	SIC #	IDC	SECURITY NAME
1	3.00	67	BTC	BELL CANADA ENTERPRISES
2	3.00	63	CBB	CHUBB CORP
3	3.00	33	X	USX CORP
4	3.00	35	HR	NAVISTAR INTL CORP
5	3.00	54	WIN	WINN DIXIE STORES INC
6	3.00	13	CBI	CBI INDS INC
7	3.00	49	WWP	WASHINGTON WTR PWR CO
8	3.00	29	AHC9	AMERADA HESS CORP
9	3.00	35	UT	UNITED TELECOMMUNICATIONS
10	3.00	10	AMX	AMAX INC
11	3.00	67	ASA	ASA LTD
12	3.00	49	PGN	PORTLAND GEN CORP
13	3.00	64	AXD	ALEXANDER & ALEXANDER SVCS
14	3.00	13	KMG	KERR MCGEE CORP
15	3.00	48	T	AMERICAN TEL & TELEG CO
16	3.00	63	CIC	CONTINENTAL CORP
17	3.00	60	FBG1	BANC ONE CORP
18	3.00	33	NS	NATIONAL INTERGROUP INC
19	3.00	26	KMB	KIMBERLY CLARK CORP
20	3.00	36	MSU9	MATSUSHITA ELEC INDL
21	3.00	13	GAS	NICOR INC
22	3.00	49	CPL	CAROLINA PWR & LT CO
23	3.00	37	UA	UNITED TECHNOLOGIES CORP
24	3.00	64	MMC	MARSH & MCLENNAN COS INC
25	3.00	12	BNI	BURLINGTON NORTHN INC
26	3.00	20	K	KELLOGG CO
27	3.00	63	UFY	USF&G CORP
28	3.00	34	AC	PRIMERICA CORP
29	3.00	13	SN	AMOCO CORP
30	3.00	27	DNY	DONNELLEY RR & SONS CO
31	2.23	10	AL	ALCAN ALUM LTD
32	1.92	33	IAD	INLAND STL INDS INC
33	1.63	32	PPG	PPG INDS INC
34	1.41	60	FML	FIRST BK SYS INC
35	1.00	49	PLT	PACIFIC LTG CORP
36	0.83	67	LEM	LEHMAN CORP
37	0.51	10	UNP	UNION PAC CORP
38	0.48	13	OXY	OCCIDENTAL PETE CGPCR

TABLE VI

SIMULATION RESULTS  
1980  
HISTORIC SIMULATION:

LIABILITY STREAM USED LONG TERM ABO

NUMBER	WEIGHT	SIC #	IDC	SECURITY NAME
1	3.00	49	NMK	NIAGARA MOHAWK PWR CORP
2	3.00	28	IFF	INTERNATIONAL FLAVORS & FR
3	3.00	60	MNK	MELLON BANK CORPORATION
4	3.00	36	WHR	WHIRLPOOL CORP
5	3.00	60	NOB	NORWEST CORP
6	3.00	26	KMB	KIMBERLY CLARK CORP
7	3.00	28	WLA	WARNER LAMBERT CO
8	3.00	20	RAL	RALSTON PURINA CO
9	3.00	49	KGE	KANSAS GAS & ELEC CO
10	3.00	28	BOR	BORG WARNER CORP
11	3.00	26	UCC	UNION CAMP CORP
12	3.00	60	CHL	CHEMICAL NEWYORK CORP
13	3.00	20	K	KELLOGG CO
14	3.00	30	GT	GOODYEAR TIRE & RUBR CO
15	3.00	49	POM	POTOMAC ELEC PWR CO
16	3.00	13	LLX	LOUISIANA LD & EXPL CO
17	3.00	28	PG	PROCTER & GAMBLE CO
18	3.00	53	FDS	FEDERATED DEPT STORES INC
19	3.00	33	NS	NATIONAL INTERGROUP INC
20	3.00	26	FHP	FORT HOWARD CORP
21	3.00	20	BN	BORDEN INC
22	3.00	20	GIS	GENERAL MLS INC
23	3.00	34	AC	PRIMERICA CORP
24	3.00	28	GLXO	GLAXO HLDGS PLC
25	3.00	34	GS	GILLETTE CO
26	3.00	37	GM	GENERAL MTRS CORP
27	3.00	23	ISS	INTERCO INC
28	3.00	48	T	AMERICAN TEL & TELEG CO
29	3.00	20	CPB	CAMPBELL SOUP CO
30	3.00	36	GNB	GOULD INC
31	2.33	37	ML	MARTIN MARIETTA CORP
32	2.06	37	F	FORD MTR CO DEL
33	2.05	36	C	CHRYSLER HLDG CO
34	1.00	49	PPL	PENNSYLVANIA PWR & LT CO
35	1.00	60	AXP	AMERICAN EXPRESS CO
36	0.95	32	OCF	OWENS CORNING FIBERGLAS CO
37	0.62	20	PSY	PILLSBURY CO

31

TABLE VII

## 1975 INDUSTRY WEIGHTINGS

Electric, Gas and Sanitation Utilities, and  
Banking Restricted to 10% of Portfolio.

Code	Industry	% of Portfolio
		5.74%
10	Metal Mining	3.00
12	Bituminous Coal & Lignite Min.	12.48
13	Oil & Gas Extraction	3.00
20	Food Kindred Products	3.00
26	Paper and Allied Products	3.00
27	Printing Publishing and A.P.	3.00
29	Petroleum Refin. & Rel. Prod.	1.63
32	Stone Clay Glass & Conc. Prod.	7.92
33	Primary Metal Industries	3.00
34	Fab. Metal Prod. Ex. M.&T.E.	6.00
35	Machinery Except Electrical	3.00
36	Ele. and Ele. Mach.	3.00
37	Transportations Equipment	3.00
48	Communication	10.00
49	Electric Gas and Sanit. Serv.	3.00
54	Food Stores	4.41
60	Banking	9.00
63	Insurance Carriers	6.00
64	Ins. Agents Brok. Serv.	6.83
67	Hold. and Other Inv. Comp.	

SPANNING PORTFOLIO  
1/1/75

Market Capitalization	(000,000)
Average	\$1,203
High	\$8,393
Low	\$ 137
Shares Outstanding	(000)
Average	74,840 shares
Dividend Yield	5.1%
Price Earning Ratio	13.1x
Number of Stocks	38
Turnover 1975-1976	21.3%

32  
TABLE VIII

## 1980 INDUSTRY WEIGHTINGS

Electric, Gas and Sanitation Utilities, and  
Banking Restricted to 10% of Portfolio.

Code	Industry	% of Portfolio
		3.00%
13	Oil	15.62
20	Food Kindred Products	3.00
23	App. & Oth. Fin. Prod.	9.00
26	Paper and Allied Products	15.00
28	Chemical and Allied Prod.	3.00
30	Rubber and Misc. Plast. Prod.	0.95
32	Stone Clay Glass & Conc. Prod.	3.00
33	Primary Metal Industries	6.00
34	Fab. Metal Prod. Ex. M.&T.E.	8.05
36	Ele. and Ele. Mach.	7.38
37	Transportations Equipment	3.00
48	Communication	10.00
49	Electric Gas and Sanit. Serv.	3.00
53	General Merch. Stores	10.00
60	Banking	

SPANNING PORTFOLIO  
1/1/80

Market Capitalization	(000,000)
Average	\$1,790
High	\$13,311
Low	\$ 247
Shares Outstanding	(000)
Average	130,229 shares
Dividend Yield	6.9%
Price Earning Ratio	6.6x
Number of Stocks	37
Turnover 1980-1981	30.4%

TABLE IX  
SIMULATION RESULTS  
1985  
HISTORIC SIMULATION:

## LIABILITY STREAM USED LONG TERM STUDY ABO

NUMBER	WEIGHT	SIC #	IDC	SECURITY NAME
1	3.00	53	JCP	PENNEY J C INC
2	3.00	49	HE	HAWAIIAN ELEC I
3	3.00	58	MCD	MCDONALDS CORP
4	3.00	60	SEK	SOUTHEAST BKG C
5	3.00	23	RML	RUSSELL CORP
6	3.00	28	AVP	AVON PRODS INC
7	3.00	60	MFT	MANUFACTURERS HAN
8	3.00	99	TBILL	CASH: TBILLS 0.0
9	3.00	49	DPL	DPL INC
10	3.00	35	MFL	MILLIPORE CORP
11	3.00	20	UL	UNILVER PLC
12	3.00	38	EK	EASTMAN KODAK C
13	3.00	27	CKH	COMMERCE CLEARI
14	3.00	28	SQB	SQUIBB CORP
15	3.00	28	PFE	PFIZER IND
16	3.00	20	KO	COCA COLA CO
17	3.00	45	DAL	DELTA AIR LINES
18	3.00	67	NES	NEW ENGLAND ELE
19	3.00	89	SW	STONE & WEBSTER
20	3.00	61	GW	GREAT WESTN FIN
21	3.00	28	SKL	SMITHKLINE BECKMAN
22	3.00	28	SRG	SCHERING PLOUGH
23	3.00	51	SY	SYSCO CORP
24	3.00	38	MM	MINNESOTA MNG &
25	2.93	78	DIS	DISNEY WALT CO
26	2.68	25	HB	HILLENBRAND IND
27	2.66	49	BGE	BALTIMORE GAS & ELE
28	2.53	38	MD	MEDTRONIC INC
29	2.30	73	KGS	KELLY SVCS INC
30	2.05	35	PR	PRIME COMPUTER
31	1.77	28	SY	SYNTEX CORP
32	1.76	16	FL	FLUOR CORP
33	1.70	60	FBG1	BANC ONE CORP
34	1.54	63	AH	AHMANSON H F &
35	1.35	49	AY	ALLEGHENY PWR SYS
36	1.22	48	CT	CENDEL CORP
37	1.09	20	CF	CPC INTL INC
38	0.77	28	PG	PROCTER & GAMBLE
39	0.59	28	AH	AMERICAN HOME PRODS
40	0.54	34	GS	GILLETTE CO
41	0.41	53	DNC	DAYTON HUDSON C
42	0.13	47	FED	FEDERAL EXPRESS

34  
TABLE X

## 1985 INDUSTRY WEIGHTINGS

Electric, Gas and Sanitation Utilities, and  
Banking Restricted to 10% of Portfolio.

Code	Industry	% of Portfolio
		1.76%
16	Const. Oth Than B.C.-G.C.	7.09
20	Food Kindred Products	3.00
23	App. & Oth. Fin. Prod.	2.68
25	Furniture and Fixtures	3.00
27	Printing Publishing and A.P.	18.13
28	Chemical and Allied Prod.	0.54
34	Fab. Metal Prod. Ex. M.&T.E.	5.05
35	Machinery Except Electrical	8.53
38	Meas. Anal. & Cont. Inst. Etc.	3.00
45	Transportation By Air	0.13
47	Transportation Services	1.22
48	Communication	10.00
49	Electric Gas and Sanit. Serv.	3.00
51	Wholesale Trade-Nondur. Goods	3.41
53	General Merch. Stores	3.00
58	Eating and Drinking Places	7.70
60	Banking	3.00
61	Cred. Agen. Oth. Than Banks	1.54
63	Insurance Carriers	3.00
67	Hold. And Other Inv. Comp.	2.30
73	Business Services	2.93
78	Motion Pictures	3.00
89	Miscellaneous Services	3.00
99	Nonclassifiable Establishments	

SPANNING PORTFOLIO  
1/1/85

Market Capitalization	(000,000)
Average	\$3,045
High	\$11,689
Low	\$ 373
Shares Outstanding	(000)
Average	97,120 shares
Dividend Yield	4.0%
Price Earning Ratio	11.0x
Number of Stocks	41
Turnover 1985-1986	15.0%



35

Table XI  
ANALYSIS OF PROGRAM SIMULATION

Target & Index represent the Standard & Poors 500 Stock Index  
portfolio represent optimizer chosen portfolio.  
One example of the benefits of the technique are seen in the  
monthly statistics, where the annualized standard deviation of  
returns is 14.6% for the optimized portfolio vs. 16.04 for the  
S&P 500.

## ANNUALIZED MONTHLY STATISTICS

	MEAN	STD DEV	SKEW * 10 <sup>6</sup>
Target	15.88%	16.04%	-50.00
Portfolio	17.97%	14.60%	-19.72
Portfolio Diff	2.09%	4.40%	.01
Index	15.88%	16.04%	-50.00
Index Diff	.00%	.00%	.00

## ANNUALIZED QUARTERLY STATISTICS

	MEAN	STD DEV	SKEW * 10 <sup>6</sup>
Target	16.18%	17.04%	-172.12
Portfolio	18.40%	16.19%	-41.29
Portfolio Diff	2.23%	3.99%	1.58
Index	16.18%	17.04%	-172.12
Index Diff	.00%	.00%	.00

## ANNUAL STATISTICS

	MEAN	STD DEV	SKEW * 10 <sup>6</sup>
Target	16.38%	13.63%	-670.98
Portfolio	19.01%	13.43%	-61.67
Portfolio Diff	2.63%	4.43%	-23.31
Index	16.38%	13.63%	-670.98
Index Diff	.00%	.00%	.00

36

statistical correlation with the liability return is further evident in Table III and FIG. 5, wherein detailed comparisons are made between the selected portfolio of assets and the standard liability return.

In FIG. 6 a range of simulation funding returns for the portfolio of assets are compared with a typical pension fund a mixture of stocks, bonds, real estate and treasury bills. Clearly, the risk is much higher for the typical pension fund; and dramatic improvement in the return, or reduction of risk, results when only 35% of the standard pension fund is modified using the method of the invention.

In another embodiment, a portfolio of assets can be constructed by selecting a portion of a total portfolio with assets having optimal correlation of asset return to a liability or financial index. The remainder of the portfolio comprises futures contracts which are combined with the correlated portfolio portion to achieve in effect an optimum correlation for the entire portfolio of assets. Further details are set forth in Appendix IV.

In another aspect of one embodiment, control can be exerted over pension plan surplus by adjusting the level of risk selected for a portfolio of assets. As illustrated in FIG. 6, the expected return can be selected at various levels with the degree of risk, or standard deviation of the funding level, generally increasing as one moves from a position of

37

minimum risk at the top of the "bullet" to higher future returns. Control over a pension plan surplus, or for that matter any plan for which you wish to respond dynamically to control risk/return in concert, can be accomplished over a wide range of risk and return values. Such an approach can be used to manage return under variable risk positions and minimize insurance costs for protecting against underfunding of a plan, such as falling below a predetermined minimum floor. Consequently, as the funding level approaches 100% a minimum risk portfolio of assets should be constructed using the methods described hereinbefore. As the surplus accumulates, the acceptable risk level can be increased for the portfolio of assets by dynamic modification of the portfolio asset components. One can utilize futures contracts as an overlay for an underlying portfolio of assets, having been selected by the basic invention described previously, to create in effect an optimum statistical correlation for the entire portfolio, including the futures contracts. As the surplus approaches 10% - 20% excess, a portfolio of assets can be constructed resulting in a much higher level of future return. For example, in FIG. 6, the change in future return from minimum risk to the highest return data point is about a 35% greater return but with an accompanying 70% - 80% increase in standard deviation compared to the minimum risk point.

38

In a further embodiment of the invention one can overcome problems associated with an indefinite covariant matrix. The current system and method does not use the full covariance matrix to extract a partial derivative to guide the search process, and thus it is not subject to one of the failures that results using standard quadratic programming.

To us standard quadratic programming algorithms the covariance matrix must be positive semi-definite, or positive definite. This means technically that no row of the matrix can be replicated by a linear combination of other rows.

However, this condition occurs when there are fewer returns than the number of securities under consideration. This is a significant shortcoming of the standard methodology. For example, to consider five hundred securities simultaneously, the user must supply at least five hundred and one returns for each security being considered. If monthly data is being used for the return series, at least forty-one years of data must be available for each security under consideration. Most securities have not been in existence for this period of time.

If not enough returns are available or if the matrix is indefinite, the standard quadratic solution techniques will fail to find a solution point that is optimal.

The current system allows a solution, which is at least as good as any other feasible solution, to be achieved without regard to the sensitivity of the covariance matrix.

39

Problems that otherwise could not be solved by standard technologies are solved by the current system. This allows practical portfolios to be selected even though there is limited available return information.

In another aspect of the invention involving practical applications of the current methodology to managing security portfolios, the ability to invest dividend income and other cash flows efficiently is also an important element in effective management. The current system allows a portfolio manager to invest available cash in the most efficient securities while considering the current portfolio holdings. Effectively this allows the manager to invest in securities which best enhance the current portfolio position without selling any of the current holdings. In a practical portfolio management system the ability to reinvest cash flows efficiently is always an important consideration. The current system provides this ability. An illustration of a reinvestment solution is provided in Appendix V which lists data used and results obtained in performing the reinvestment method.

In addition to finding optimal portfolios for tracking financial targets, the current system allows other objectives to be considered and incorporated in the solution. Examples include supplementing the basic objective function with an income objective, tax impact objective, or a company cash flow

40

objective. The system is flexible enough to allow the objective function to be customized for particular applications.

As an example, an investor who has a preference for securities with high dividend yields will specify an objective function which explicitly states the trade off between portfolio tracking and dividend income. The computer routine to optimize a portfolio (the optimizer) will extract the partial derivatives of this objective function and proceed to select an optimal portfolio which exactly incorporates the specified trade-off between dividend income and tracking. This investor then has a custom solution which addresses his particular concerns and requirements.

The consideration of transaction costs associated with buying and selling securities is incorporated into the portfolio optimization system to control expenses due to trading. The trade off between tracking accuracy and transaction costs can be specified by the investor. This allows for a customized objective function, with regard to expected transaction costs, for each client.

Investors may also have tax effects to consider when trading from one security to another. The taxable gains and losses and their impact on expected return can be specified by the investor so as to control these costs.

Another preference which can be incorporated into the objective function and handled explicitly by the optimization

4/

system is the consideration of cash flows. An investor may have a preference for investing in companies which have large and positive cash flows. This objective can be incorporated into the system and resulting portfolios will reflect this investment goal.

While preferred embodiments of the present invention have been illustrated and described, it will be understood that changes and modifications can be made therein without departing from the invention to its broader aspects. Various features of the invention are defined in the following claims.

42

APPENDICES TO SPECIFICATION

- APPENDIX I - Spanning Program for creating an optimal target tracking portfolio of securities (Appendix I pp. 1-27). Also includes 4 pages of Output.
- APPENDIX II - Liability Return Program for creating a return series for a liability stream from yields (Appendix II pp. 1-2).
- APPENDIX III - Correlation Portfolio Program for creating an optimal index correlation portfolio with securities (Appendix III pp. 1-30).
- APPENDIX IV - "Fasttrack" Program for analyzing large numbers of securities in a rapid, efficient manner to provide optimum correlation of asset return to a time dependent financial index. Substantial computer memory storage reduction is also achieved.
- APPENDIX V - Data listing of method for reinvestment of available cash flow beginning from a current portfolio.
- APPENDIX VI - Example illustrating distinctions between current system and prior art.



43

APPENDIX I

SPANNING PROGRAM

To create an optimal target tracking portfolio of securities.

(Appendix I pp. 1- 27)

```

      TURN = XNUM*(100./(TURN1+.01))-100.0/100.01)
    END IF
    ***** CALL COVIN TO READ IN RETURNS *****
    CALL COVIN (NUM, IREAD, IFIRST, ILAST)
    ***** CALL HOLDING TO FOR CURRENT WEIGHTS *****
    CALL HOLDING (NUM, HOLD, IPORT)
    ***** CALL MATRIX TO CALCULATE INPUT ARRAYS *****
    WRITE(*,101)'CALLING MATRIX .....
    CALL MATRIX (NUM, IREAD)
    ***** CALLING BOUNDS *****
    CALL BOUNDS (NUM, ICOL, NMM)
    ***** CALL TO OPTIMIZER *****
    WRITE(*,101)'CALLING OPTIMIZER .....
    IF (ICOUNT.EQV..TRUE.) THEN
    END IF
    MAXHS = *(2*NUM+1D)*(2*NUM+1D)+8*(2*NUM+1D)+6
    ITURN = 1
    CALL QSET ( NUM, MAXHS, ITURN )
    ***** IF TURNOVER IS CONSIDERED, CALL TURNOV *****
    IF (TURN .GT. 0.0) THEN
      ITURN = 1
      CALL QSET (NUM, MAXHS, ITURN)
    END IF
    ***** CALL TO PORTFOLIO OUTPUT *****
    WRITE(*,101)'CALLING PORT .....
    CALL PORT (NUM)
    IF (IDUMPY.EQ. 1) GO TO 100
    ***** CALL TO STATISTICAL TRACKING *****
    CALL TRACKER (NUM, IREAD)
    ***** CALL TO TURNOVER ROUTINE *****
    WRITE(*,101)'CALLING BAL .....
    CALL BAL (NUM)
    ***** SPANNING SIMULATION COMPLETE *****
    30 WRITE(*,101)'SPANNING SIMULATION COMPLETE'

      IF (HOLDX.EQV..FALSE.) THEN
        WRITE(*,101)'FILE NOT FOUND ', HOLD
        GO TO 11
      END IF
      ***** ICOUNT IS TRUE --> CALL MULTRUN *****
      END IF
      ELSE
        WRITE(*,101)'CALLING MULTRUN .....
        CALL MULTRUN(ISKIP)
      END IF
      ***** CALL SAVDAT *****
      CALL SAVDAT
      ***** CALL SAVDAT TO SAVE PAST.DAT *****
      *****
      NM = NMSAVE
      NM = (INT(NM/100)-80)*12+(NM-INT(NM/100))*100
      IREAD = NSTAT+NSIMS
      IF (NSTOCKS.GT.0) THEN
        XBUPPER = SMAX
      END IF
      ***** SET UP CONSTANTS *****
      NUM = NSTOCKS+NBONDS
      IF (NUM.GT.ND) THEN
        WRITE(*,101)'WARNING NUM > ', ND, ' TOO BIG FOR PROGRAM'
        STOP
      END IF
      ICOL = 1D
      KE = KEQ
      KKK = KE
      IA = ND
      ICC = ND
      IH = N2D
      IFIRST = NM-NSTAT
      ILAST = NM+NSIMS-1
      BULL1 = BULLET
      TARG1 = TARGET
      TURN1 = TURN
      BULLET = BULLET*ABS(BULLET)/1000.
      IF (TURN1.EQ.100.) THEN
        TURN = 0.0
      ELSE
        WRITE(*,101)'INPUT NUMERATOR'
        READ(*,101)XNUM
        XNUM = 0.10
      END IF
    C
  C

```

44

45

```

C UPDATE December 6, 1988
C *****
  PROGRAM SPAN
C *****

  INCLUDE 'COMMON.F'
  CHARACTER*30 HOLD
  LOGICAL ICOUNT
  LOGICAL HOLDEX

C ***** WRITE COPYRIGHT TO SCREEN *****
  INCLUDE 'COPYRIGHT.F'
C *****

C ***** DATA INPUT SECTION *****
5 CONTINUE
C DEFAULT READ IN OF 24 MONTHS PRIOR = NSTAT, NRESTS FOR B( ) ****
  NSTAT = 24
  NRETS = 24
  BULLET = 0.0
  TARGET = 1.
  TURN = 100.
  ITYPE = 1
  ICHANGE = 0
  XBLOWER = .000
  XBUPPER = 3.000
  SMAX = XBUPPER

C ***** CALL FLASH TO INTRODUCE S P A N *****
C CALL FLASH

C ***** IF 'COUNTS' EXISTS THIS IS A MULTIPLE RUN *****
  INQUIRE (FILE = 'COUNTS', EXIST = ICOUNT)

  IF (ICOUNT .EQV. .FALSE.) THEN
C ***** CALL INTRO TO INPUT PARAMETERS *****
    CALL INTRO
C ***** CALL MODIFY TO CHANGE PARAMETERS *****
    CALL MODIFY
C ***** READ IN THE HOLDINGS FILE *****
    IF (IHOLD .NE. 0) THEN
11 WRITE(*,*) 'INPUT PORTFOLIO #, AND NAME OF HOLDINGS FILE'
    READ(*,*) IPORT, HOLD
    INQUIRE (FILE = HOLD, EXIST = HOLDEX)

```

46

```

CLOSE(13)
RETURN

C ***** STOP DONE WITH MULTI RUN *****
99 CLOSE(13)
WRITE(*,*) 'DONE WITH MULTIPLE RUN *****'
STOP
END

***** END OF SUBROUTINE MULTRUN *****
***** SUBROUTINE HOLDING (NUM_HOLD,IPORT) *****
***** INCLUDE 'COMMON.F' *****
***** DIMENSION VAL(ND) *****
***** CHARACTER*9 SYMBL, IDC SYN(ND) *****
***** CHARACTER*30 HOLD *****

C ***** I HOLD IS READ FROM FIRST SCREEN RESPONSE: OLD OR HOLDINGS RUN **
IF (HOLD.EQ. 0) GO TO 999

OPEN (25, FILE = HOLD, STATUS = 'OLD')

IPORT = REAL(IPORT)
- 0
READ(25,*, END = 44)SYMBL,VALUE
IF (SYMBL.EQ. 'IDCSYMBL') THEN
  IF (VALUE.EQ. XPORT) THEN
    READ(25,*, END = 44)GARB
    READ(25,*, END = 44)SYMBL,VALUE
    IF (SYMBL.EQ. 'IDCSYMBL') GO TO 44
  - 1 + 1
  IDC SYN(1) = SYMBL
  VAL(1) = VALUE
  GO TO 33
END IF
READ(25,*, END = 44) GARB
END IF
GO TO 22

44 CONTINUE
NIDC = 1
IF (NIDC.EQ. 0) THEN
  WRITE(*,*) 'PORTFOLIO NUMBER NOT FOUND', IPORT
  WRITE(*,*) 'ENTER PORTFOLIO NUMBER AGAIN'
  READ(*,*) IPORT
  REWIND(25)
  GO TO 11
END IF

IF (ICOUNT.EQV. .TRUE.) THEN
  WRITE(*,102) 'DONE WITH RUN NUMBER', ISKIP
  GO TO 5
END IF

101 FORMAT(*,A50)
102 FORMAT(*,A50,17)

END

***** END OF MAIN PROGRAM SPAN *****
***** SUBROUTINE MULTRUN (ISKIP) *****
***** INCLUDE 'COMMON.F' *****
***** CHARACTER*30 XTITL,XFLOUT,XLIBIN,XSTKIN,XBNDIN *****
***** OPEN (13, FILE = 'MULTIN', STATUS = 'OLD') *****

IF (ISKIP.EQ. 0) GO TO 30

DO 10 I = 1, ISKIP
  DO 20 J = 1, 18
    READ(13, '(A1)', END = 99)GARB
    CONTINUE
  CONTINUE
  CONTINUE

  READ(13,101, END = 99)TITL
  READ(13,101, END = 99) FILEOUT
  READ(13,101, END = 99) LIBIN
  READ(13,101, END = 99) STOCKIN
  READ(13,101, END = 99) BONDIN
  READ(13,*, END = 99) XMSAVE
  READ(13,*, END = 99) XNSINS
  READ(13,*, END = 99) XNSTOCKS
  READ(13,*, END = 99) XNBONDS
  READ(13,*, END = 99) XIDURMY
  READ(13,*, END = 99) XSTOCKMIN
  READ(13,*, END = 99) XIMAX
  READ(13,*, END = 99) XSMAX
  READ(13,*, END = 99) XNSTAT
  READ(13,*, END = 99) XNRETS
  READ(13,*, END = 99) XBULLET
  READ(13,*, END = 99) XTARGET
  READ(13,*, END = 99) XTURNS

  101 FORMAT (A20)
  ISKIP = ISKIP+1

```

47

```

10 ICASH = 0
PORTVAL = 0.0
DO 10 I=1,NIDC
PORTVAL = PORTVAL + VAL(I)
IF (IDCSYM(I) .EQ. 'CASH') THEN
TEMP = VAL(I)
VAL(I) = VAL(I)
VAL(I) = TEMP
GARB = IDCSYM(I)
IDCSYM(I) = IDCSYM(I)
IDCSYM(I) = GARB
END IF
CONTINUE
IF (ICASH .EQ. 0) THEN
WRITE(*,*) 'NO CASH INCLUDED.....PROGRAM STOPPING'
STOP
END IF

10 ITEST = 0
ICOUNT = 0
DO 20 I = 2,NIDC
ITEST = 0
DO 30 J = 1,NUM
IF (IDC(J)(1:4) .EQ. IDCSYM(I)) THEN
ITEST = 1
NCOUNT = NCOUNT+1
OLDWT(J) = VAL(I) / PORTVAL
END IF
CONTINUE
IF (ITEST .EQ. 0) THEN
WRITE(*,*) IDCSYM(I), ': SYMBOL NOT INCLUDED IN RETURN FILE'
END IF
CONTINUE

20 IF (IDCTEST .EQ. 1) THEN
WRITE(*,*) NIDC-NCOUNT-1, 'SYMBOLS NOT FOUND...PROGRAM STOPPING'
STOP
EN) IF
RETURN

999 OPEN(21, FILE = 'OLD', STATUS = 'OLD')
DO 90 I = 1,NUM
READ(21,*) OLDWT(I)
IF (TURN .EQ. 0.0) OLDWT(I) = 0.0
OLDWT(I) = OLDWT(I)/100.
CONTINUE
CLOSE(21)
RET JRN

90

```

```

*****
C ***** END OF SUBROUTINE HOLDING *****
C UPDATE December 6, 1988
C ***** THIS SUBROUTINE READS IN STOCK, BOND & LIA DATA *****
SUBROUTINE COVIN (NUM, IREAD, IFIRST, ILAST)
C *****
C IFIRST IS FIRST MONTH OF DATA NEEDED FOR MATRIX
C ILAST IS LAST MONTH OF SIMULATION
C INCLUDE 'COMMON.F'
C ***** READ IN LIABILITY RETURNS *****
OPEN (9, FILE = 'LIABIN, STATUS = 'OLD')
READ (9,111) LIANAME
WRITE(*,*) LIANAME
DO 200 I = 1,5
READ (9,111) GARB
200 CONTINUE
READ (9,*) ISTART
READ (9,*) ISTOP
DO 300 I = 1, IFIRST-ISTART
READ (9,111) GARB
300 CONTINUE
DO 400 I = 1, IREAD
READ (9,*) ANIM(I)
400 CONTINUE
CLOSE(9)
IF (NSTOCKS.GT.0) THEN
OPEN (4, FILE = 'STOCKIN, STATUS = 'OLD')
DO 600 I = 1, ITOUPLY
READ (4,111) (GARB,L = 1,6)
READ (4,*) ISTART
READ (4,*) ISTOP
DO 610 K = 1, ISTOP-ISTART+1
READ (4,*) XGARB
CONTINUE
FORMAT (A40)
610
600
111
DO 700 I = 1, NSTOCKS

```

48

```

783      READ (4,111,END = 785) NAME(I)
C ***** OUT OFF LEADING BLANKS OF NAME(I)
      ILEN = 1
      GARB = NAME(I)
790      IF (NAME(I)(ILEN:ILEN).EQ.' ') THEN
          GO TO 790
      ELSE
          GARB(1:ILEN) = NAME(I)(ILEN:ILEN)
          END IF
          NAME(I) = GARB
          WRITE (*,112) I,NAME(I)
          FORMAT('STOCK #',15,' IS ',A40)
          READ (4,111) GARB
          GO TO 790
112      C ***** OUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER
          IDC(I) = 1
          IF (GARB(ILEN:ILEN).EQ.' ') THEN
              ILEN = ILEN + 1
              IF ((ILEN+3).GT.LEN) GOTO 793
              GOTO 791
          END IF
          IDC(I)(1:4) = GARB(ILEN:ILEN+3)
          ILEN = ILEN + 4
792      IF (GARB(ILEN:ILEN).EQ.' ') THEN
              ILEN = ILEN + 1
              IF ((ILEN+3).GT.LEN) GOTO 793
              GOTO 791
          END IF
          IDC(I)(6:9) = GARB(ILEN:ILEN+3)
793      CONTINUE

      READ (4,*) ISIC(I)
      IF (ISIC(I).GT.99) ISIC(I) = INT (ISIC(I)/100)
      READ (4,111) GARB,L = 4,5)
      READ (4,*) ISTART
      READ (4,*) ISTOP

C ***** TEST FOR ENOUGH RETURNS TO SATISFY NRETS *****
      ICHECK = NM-NRETS-ISTART
      IF (ICHECK.LT.0) NRETS = NM - ISTART
      IF (NRETS.LT.NSTAT) ICHECK = IFIRST-ISTART
      DO 800 J = 1,ICHECK

800      READ (4,111) GARB
          CONTINUE
          DO 801 J = 1,NRETS-NSTAT
              READ(4,*)GARB
              CONTINUE
          JJ = NRETS-NSTAT
          DO 810 K = 1,IREAD
              READ (4,*) RET(K,1)
              ***** ADJUST OCTOBER 1987 *****
              IF (K-NSTAT-1.EQ. 94) THEN
                  IF (K.LE.NSTAT) RET(K,1)=RET(K,1)/2.97
              END IF
              J = JJ+K
          CONTINUE
          DO 1200 J = 1,ISTOP-ILAST
              READ (4,111) GARB
              CONTINUE
          CONTINUE
          ENDIF
          CONTINUE
          CLOSE(4)
          C
          IF (NBONDS.GT.0) THEN
              OPEN (7,FILE = BONDIN, STATUS = 'OLD')
              DO 900 I = NSTOCKS+1,NSTOCKS+NBONDS
                  READ (7,111) NAME(I)
                  WRITE (*,113) I,NAME(I)
                  READ (7,111) (GARB,L = 1,5)
                  READ (7,*) ISTART
                  READ (7,*) ISTOP
                  FORMAT(1,*,BOND #',15,' IS ',A40)
113      C ***** TEST ICHECK FOR ENOUGH RETURNS FOR BRET *****
                  ICHECK = NM-NRETS-ISTART
                  IF (ICHECK.LT.0) THEN
                      WRITE(*,*)BOND',1,' DOES NOT HAVE ENOUGH RETURNS FOR NRETS'
                      STOP
                  END IF
                  IF (NRETS.LT.NSTAT) ICHECK = IFIRST-ISTART
                  DO 1000 J = 1,ICHECK
                      READ (7,111) GARB
                      CONTINUE
1000

```

```
IF (ISIC(1).EQ.48) SIC49 = SIC49+OLDWT(1)
IF (ISIC(1).EQ.49) SIC49 = SIC49+OLDWT(1)
IF (ISIC(1).EQ.60) SIC60 = SIC60+OLDWT(1)
IF (ISIC(1).EQ.63) SIC63 = SIC63+OLDWT(1)
```

56



51

```

WRITE(10,*)
C ***** CALCULATE THE PORTFOLIO RETURNS *****
C ***** FIRST CALCULATE RETURNS FOR NSTAT PERIOD *****
DO 800 I = 1,NSTAT
  PRET(1) = 0.0
  DO 900 J = 1,NUM
    PRET(1,J) = PRET(1) + RET(1,J)*X(J)
  CONTINUE
800 CONTINUE
C ***** NOW CALCULATE RETURNS AND WEIGHTS IN SIMULATION PERIOD *****
DO 801 I = 1,NSIMS
  PRET(I+NSTAT) = 0.0
  DO 901 J = 1,NUM
    PRET(I+NSTAT,J) = PRET(I+NSTAT) + CUM(J)*RET(I+NSTAT,J)
  CONTINUE
901 DO 1001 K = 1,NUM
  CUM(K) = CUM(K) + (1.+RET(I+NSTAT,K))/(1.+PRET(I+NSTAT))
1001 CONTINUE
801 CONTINUE
C *** WRITE OUT THE NEW WEIGHTS TO A FILE CALLED 'OLD' *****
OPEN(15,FILE = 'OLD', STATUS = 'OLD')
REWIND(15)
DO 140 I = 1,NUM
  WRITE(15,940) CUM(I)*100
140 CONTINUE
940 FORMAT(1X,F15.8)
CLOSE(15)
C ***** CALCULATE PARTIALS AND WRITE OUTPUT *****
XVAR = 0.0
XRET = 0.0
ICOUNT = 0
DO 131 I = 1,NUM
  C(I,1) = 0.0
  DO 132 J = 1,NUM
    C(I,1) = C(I,1) + X(J)*A(I,J)*2
    XVAR = XVAR + X(I)*X(J)*A(I,J)
  CONTINUE
132 ICOUNT = ICOUNT + 1
IF(1.GT.NSTOCKS) ICOUNT = 1
WRITE(10,16) ICOUNT, X(1)*100, CUM(1)*100, C(1,1)*100,
  & ISIC(1), IDC(1), NAME(1)
  &
  & (RET1 = AVE(1) + UANIM
  & (RET = XRET + X(1)*XRET1
131 CONTINUE
16 FORMAT(1X,14,1X,2F6.3,F7.3,14,1X,A9,1X,A36)

```

```

WRITE(10,*)
STD = (XVAR*12)**(0.5)*100
XRET = XRET*12.0*100.0
649 WRITE(10,649) STD, XRET
FORMAT(1X,' MINIMUM STD DEV =',F10.4,' EXPECTED RETURN =',F10.4)
WRITE(10,*)
C IF(ITYPE.EQ.1.AND.XTOT.LT..995) WRITE(10,*) ' ERROR ***** ',
C . SUM OF X S = %,XTOT*100.
C IF(ITYPE.EQ.1.AND.XTOT.GT.1.005) WRITE(10,*) ' ERROR ***** ',
C . SUM OF X S = %,XTOT*100.
RETURN
END
C ***** END OF SUBROUTINE PORT *****
C ***** SUBROUTINE TRACKER (NUM,IREAD) *****
C ***** INCLUDE 'COMMON.F' *****
C ***** SET AVERAGES AND COVARIANCES TO ZERO *****
DO 400 I = 1,3
  AVE(I) = 0.0
  DO 480 J = 1,3
    COV(I,J) = 0.0
  CONTINUE
400 CONTINUE
C *** SET STAT ARRAY 1 = LIABILITY, 2 = PORTFOLIO, 3 = DIFFERENCES **
DO 250 I = 1,IREAD
  STAT(1,1) = ANIM(1)
  STAT(1,2) = PRET(1)
  STAT(1,3) = (PRET(1) - ANIM(1))
250 CONTINUE
C ***** GO AROUND STATISTICS IF NSIM < 3
IF(NSIMS.LT.3) GOTO 318
C ***** CALCULATE AVERAGES *****
C ***** SET NSIM TO 2 FOR STATISTICS IF NSIMS.LT.1 *****
NSIM = NSIMS
IF(NSIMS.LT.1) NSIM = 2
DO 260 I = NSTAT + 1, IREAD
  AVE(1) = AVE(1) + ANIM(1)/DBLE(NSIM)

```

52

```

260 C *****
      AVE(2)=-AVE(2)+PRET(1)/DBLE(NSIM)
      AVE(3)=-AVE(3)+(PRET(1)-ANIM(1))/DBLE(NSIM)
      C *****
      C ***** CALCULATE COVARIANCES *****
      DO 950 I=1,3
        VAR(I)=0.0
        DO 950 K=NSTAT+1,IREAD
          VAR(I)=VAR(I)+(STAT(K,I)-AVE(1))**2/DBLE(NSIM-1)
        CONTINUE
      950 C *****

      C ***** CALCULATE THE COVARIANCE MATRIX'S UPPER OFF-DIAGONAL *****
      DO 960 J=1,3
        DO 960 I=1,J-1
          DO 828 K=NSTAT+1,IREAD
            COV(I,J)=COV(I,J)+(STAT(K,I)-AVE(1))*(STAT(K,J)-AVE(J))
          CONTINUE
          COV(I,J)=COV(I,J)/DBLE(NSIM-1)
          COV(J,I)=COV(I,J)
        CONTINUE
      960 C *****

      C ***** CALCULATE CORRELATIONS *****
      DO 951 I=1,3
        DO 951 J=1,I-1
          IF (VAR(I).LE.0.0 OR VAR(J).LE.0.0) GO TO 951
          COR(I,J)=COV(I,J)/(VAR(I)**.5*VAR(J)**.5)
        CONTINUE
      951 C *****

      C ***** CALCULATE CUMULATIVE RETURNS *****
      PRET(NSTAT)=100
      ANIM(NSTAT)=100
      DO 559 I=NSTAT+1,IREAD
        X=PRET(I)
        Y=ANIM(I)
        PRET(I)=PRET(I-1)*(1+X)
        ANIM(I)=ANIM(I-1)*(1+Y)
      CONTINUE
      559 C *****

      IF (VAR(1).NE.0.0) BETA = COV(1,2)/VAR(1)
      ALPHA = AVE(2) - BETA*AVE(1)

      SST = 0.0
      SSE = 0.0
      DO 552 I = NSTAT+1,IREAD
        SSE = SSE + (ALPHA+BETA*STAT(I,1)-STAT(I,2))**2
        SST = SST + (STAT(I,2)-AVE(2))**2
      CONTINUE
      552 C *****

      AVE(2)=-AVE(2)+PRET(1)/DBLE(NSIMS-2)**.5
      IF (SST.NE.0.0) RSQ = 1 - SSE/SST
      C *****
      C ***** BEGIN WRITING THE TRACKER OUTPUT *****
      318 WRITE(10,*) ' TRACKER OUTPUT FILE '
      WRITE(10,*)
      WRITE(10,*) ' MONTH LIA PORT DIF '
      WRITE(10,*)
      C *****
      C ***** WRITE MONTH, LIABILITY RET, PORTFOLIO RET, DIFFERENCE *****
      DO 700 I=NSTAT+1,IREAD
        ISTAT = I-(NSTAT+1)
        WRITE(10,612) ISTAT+NM, STAT(I,J), J=1,3)
        FORMAT(1X,3X,14,3F13.10)
      612 CONTINUE
      700 C *****

      C ***** WRITE CUMULATIVE RETURNS *****
      WRITE(10,*)
      WRITE(10,886)
      FORMAT(1X, ' CUMULATIVE VALUES OF LIABILITIES AND ASSETS' /,
        /, ' MONTH LIABILITIES PORTFOLIO' /,
        /)
      DO 847 I=NSTAT,IREAD
        ISTAT = I-NSTAT-1
        WRITE(10,328) ISTAT+NM, ANIM(I), PRET(I)
        FORMAT(1X,15,7X,F7.3,10X,F7.3)
      328 CONTINUE
      847 CONTINUE
      837 CONTINUE

      C ***** WRITE STATISTICS BASED ON SIMULATIONS MONTHS *****
      WRITE(10,*)
      WRITE(10,*) '*****'
      WRITE(10,985) NM, NM+NSIMS-1
      985 FORMAT(/, ' STATISTICS BASED ON MONTHS ',14,' THROUGH',14)

      C ***** WRITE ALPHA, BETA, STDERR, CORREL, AND R-SQUARED *****
      WRITE(10,*)
      WRITE(10,987) 'ALPHA', ALPHA, 'ANNUAL', ALPHA*1200
      WRITE(10,989) 'BETA', BETA
      WRITE(10,988) 'STD ERR', STDERR*100
      WRITE(10,988) 'CORRELATION', CORREL(2,1)
      WRITE(10,988) 'R-SQUARED', RSQ*100
      987 FORMAT(1X,A12.10X,F10.6,8X,F10.2,' %')
      988
      989 FORMAT(1X,A12.10X,F10.6,8X,F10.2,' %')

```

53

```

C ***** WRITE AVERAGES AND VARIANCES OF 1-LIAB, 2-PORT, 3-DIFFS **
WRITE(10,*)
WRITE(10,*)
WRITE(10,*) ANNUAL RETURNS:
WRITE(10,*) MEAN
WRITE(10,*) STD DEV
DO 990 I = 1,3
  AVE(I) = (AVE(I)*12)*100
  VAR(I) = ((VAR(I)*12)**.5)*100
CONTINUE
990
WRITE(10,916) 'LIABILITY', AVE(1), VAR(1)
WRITE(10,916) 'PORTFOLIO', AVE(2), VAR(2)
WRITE(10,916) 'DIFFERENCE', AVE(3), VAR(3)
916 FORMAT(1X,3X,A11,9X,F9.2,*,10X,F9.2,*,*)
C ***** WRITE PARAMETERS FOR THE RUN *****
WRITE(10,*)
WRITE(10,*)
WRITE(10,*) PARAMETERS FOR THIS RUN WERE:
WRITE(10,*)
WRITE(10,*) TITLE
WRITE(10,102) FILEOUT
WRITE(10,103) LIABIN
WRITE(10,104) STOCKIN
WRITE(10,105) BONDIN
WRITE(10,106) NRSAVE
WRITE(10,107) NSINS
WRITE(10,108) NSTOCKS
WRITE(10,109) NBONDS
WRITE(10,110) IDUMMY
WRITE(10,111) STOCKMIN
WRITE(10,112) YIMAX
WRITE(10,113) SMAX
WRITE(10,114) NSTAT
WRITE(10,115) NRETS
WRITE(10,116) BULL1
WRITE(10,117) TARG1
WRITE(10,118) TURN1
601 FORMAT(1X,*,1) SPANNING RUN TITLE
602 FORMAT(1X,*,2) OUTPUT FILENAME
603 FORMAT(1X,*,3) LIABILITY STREAM FILE
604 FORMAT(1X,*,4) STOCK RETURN FILE
605 FORMAT(1X,*,5) BOND RETURN FILE
606 FORMAT(1X,*,6) MONTH SIMULATION BEGINS
607 FORMAT(1X,*,7) NUMBER OF MONTHS SIMULATED
608 FORMAT(1X,*,8) NUMBER OF STOCKS
609 FORMAT(1X,*,9) NUMBER OF BONDS
610 FORMAT(1X,*,10) STOCKS TO READ PAST
C ***** WRITE HISTORICAL DATA USED TO MAKE RUN *****
WRITE(10,*)
WRITE(10,*) PORTFOLIO CONSTRUCTED UPON THE FOLLOWING DATA:
WRITE(10,*)
WRITE(10,*) MONTH LIA PORT DIF
DO 300 I=1,NSTAT
  WRITE(10,612) I,NH-(NSTAT+1),(STAT(1,J),J=1,3)
300 CONTINUE
END
RETURN
END
***** END OF SUBROUTINE TRACKER *****
SUBROUTINE BAL (NUM)
*****
INCLUDE 'COMMON.F'
DIMENSION CC(ND,2)
INTEGER BUY(ND), SELL(ND)
VNEW = 0.0
VOLD = 0.0
DO 10 I=1,NUM
  CC(I,1) = 0.0
  CC(I,2) = 0.0
  DO 20 J=1,NUM
    CC(I,J) = CC(I,1)+X(J)*A(I,J)*2.
    VNEW = VNEW+X(I)*X(J)*A(I,J)
  CC(I,2) = CC(I,2)+OLDWT(J)*A(I,J)*2.
  VOLD = VOLD+OLDWT(I)*OLDWT(J)*A(I,J)
20 CONTINUE
CONTINUE
***** DELTA IS THE PERCENTAGE IMPROVEMENT IN VARIANCE *****
DELTA = (VOLD - VNEW)/VNEW
DELTA = DELTA * 10000.
***** IF THE OLD VARIANCE IS ZERO USE THE NEW PARTIALS TO SORT **
KVAR = 2
IF (VOLD.EQ.0) KVAR = 1
IBUY = 0

```





56

57

58

```

204 ELSE IF (IRESPON.EQ.4) THEN
    WRITE(*,124)
    READ(*,*,END = 204, ERR = 204) STOCKIN
205 ELSE IF (IRESPON.EQ.5) THEN
    WRITE(*,125)
    READ(*,*,END = 205, ERR = 205) BONDIN
206 ELSE IF (IRESPON.EQ.6) THEN
    WRITE(*,126)
    READ(*,*,END = 206, ERR = 206) MNSAVE
207 ELSE IF (IRESPON.EQ.7) THEN
    WRITE(*,127)
    READ(*,*,END = 207, ERR = 207) NSIMS
208 ELSE IF (IRESPON.EQ.8) THEN
    WRITE(*,128)
    READ(*,*,END = 208, ERR = 208) NSTOCKS
209 ELSE IF (IRESPON.EQ.9) THEN
    WRITE(*,129)
    READ(*,*,END = 209, ERR = 209) NBONDS
210 ELSE IF (IRESPON.EQ.10) THEN
    WRITE(*,130)
    READ(*,*,END = 210, ERR = 210) IDUMMY
211 ELSE IF (IRESPON.EQ.11) THEN
    WRITE(*,131)
    READ(*,*,END = 211, ERR = 211) STOCKMIN
212 ELSE IF (IRESPON.EQ.12) THEN
    WRITE(*,132)
    READ(*,*,END = 212, ERR = 212) YIMAX
213 ELSE IF (IRESPON.EQ.13) THEN
    WRITE(*,133)
    READ(*,*,END = 213, ERR = 213) SMAX
214 ELSE IF (IRESPON.EQ.14) THEN
    WRITE(*,134)
    READ(*,*,END = 214, ERR = 214) NSTAT
215 ELSE IF (IRESPON.EQ.15) THEN
    WRITE(*,135)
    READ(*,*,END = 215, ERR = 215) MRETS
216 ELSE IF (IRESPON.EQ.16) THEN
    WRITE(*,136)
    READ(*,*,END = 216, ERR = 216) BULLET

    READ(*,*,END = 10, ERR = 10) IDUMMY
    WRITE(*,*)
    WRITE(*,111)
    FORMAT(111) ENTER MINIMUM STOCK HOLDING - % , $)
    READ(*,*,END = 11, ERR = 11) STOCKMIN
    WRITE(*,*)
    WRITE(*,112)
    FORMAT(112) ENTER MAXIMUM INDUSTRY HOLDING - % , $)
    READ(*,*,END = 12, ERR = 12) YIMAX
    WRITE(*,*)
    RETURN
100 FORMAT(A35)
***** END OF SUBROUTINE REDSCRN *****
***** SUBROUTINE MODIFY *****
***** THIS ROUTINE MODIFIES THE INPUT DATA IF NECESSARY *****
***** INCLUDE 'COMMON.F' *****
10 WRITE(*,100)(' ', I = 1,5)
***** WRITE(*,*)***** CURRENTLY SELECTED PARAMETERS *****
***** CALL DISPLAY ***** DISPLAY SELECTED PARAMETERS *****
    IRESPON = 0
    WRITE(*,120)
    READ(*,*,END = 20, ERR = 10) IRESPON
    IF (IRESPON.EQ.0) THEN
        RETURN
    ELSE IF (IRESPON.EQ.1) THEN
        WRITE(*,121)
        READ(*,*,END = 201, ERR = 201) TITLE
201
    ELSE IF (IRESPON.EQ.2) THEN
        WRITE(*,122)
        READ(*,*,END = 202, ERR = 202) FILEOUT
202
    ELSE IF (IRESPON.EQ.3) THEN
        WRITE(*,123)
        READ(*,*,END = 203, ERR = 203) LIABIN
203

```



59

```

217 ELSE IF (IRESPO.NEQ.17) THEN
      WRITE(*,137)
      READ(*,*,END = 217, ERR = 217) TARGET
218 ELSE IF (IRESPO.NEQ.18) THEN
      WRITE(*,138)
      READ(*,*,END = 218, ERR = 218) TURN
      END IF
      GO TO 10
20 RETURN
100 FORMAT('O',A1)
120 FORMAT('O',*,A1)
121 FORMAT('O',*,A1)
122 FORMAT('O',*,A1)
123 FORMAT('O',*,A1)
124 FORMAT('O',*,A1)
125 FORMAT('O',*,A1)
126 FORMAT('O',*,A1)
127 FORMAT('O',*,A1)
128 FORMAT('O',*,A1)
129 FORMAT('O',*,A1)
130 FORMAT('O',*,A1)
131 FORMAT('O',*,A1)
132 FORMAT('O',*,A1)
133 FORMAT('O',*,A1)
134 FORMAT('O',*,A1)
135 FORMAT('O',*,A1)
136 FORMAT('O',*,A1)
137 FORMAT('O',*,A1)
138 END
*****
***** END OF SUBROUTINE MODIFY *****
*****
***** SUBROUTINE DISPLAY *****
*****
***** THIS ROUTINE DISPLAYS SELECTED PARAMETERS FOR SPANNING RUN *****
*****
***** INCLUDE 'COMMON.F' *****
*****
***** WRITE(*,100)
***** WRITE(*,101) TITLE
***** WRITE(*,102) FILEOUT
***** WRITE(*,103) LIABIN
***** WRITE(*,104) STOCKIN

```

60

3

62

```

C      L1 = LEN(PFILE)
      L1 = 30
      L2 = 0
      L3 = 0
      DO 30 I = 1, L1
        IF (PFILE(I:1) .EQ. ' ') GO TO 30
        L2 = L2 + 1
      PFILES(L2:L2) = PFILE(I:1)
    30 CONTINUE
      IF (L1 .GT. 0) GO TO 40
      BLANK FILE NAME, DISCARD OUTPUT TO BE FILED.
      JDATA = 0
      JPIVOT = 0
      IF (JOUT .EQ. 2) GO TO 90
      JIT = 0
      JSOL = 0
      GO TO 50
    C      OPEN PRINT FILE.
    C      CONTINUE
    C      OPEN (LUPFILE, FILE=PFILES, STATUS='UNKNOWN', ERR=60)
    C      OPEN (LUFILF, FILE=PFILES)
    C      L03 = LUFILF
    C      IF (JOUT .EQ. 1) GO TO 50
    C      L02 = LUFILF
    C      GO TO 30
    C      CONTINUE
    C      L02 = .JPRINT
    C      GO TO 10
    C      CANNOT OPEN FILE. MAKE ONE UP.
    C      CONTINUE
    C      IF (L3 .EQ. 1) GO TO 76
    C      L3 = 1
    C      WRITE (L01, 70) NCALL, PFILES(1:12)
    C      FORMAT ( //, // 5X, 'QUADPR CALL', 14
    C      1 // 1X, '*** UNABLE TO OPEN THE FILE ', A)
    C      WRITE (PFILES, 72) NCALL
    C      FORM FILE NAME.
    C      72 FORMAT ('ODPR', 14 A, ' LIS', 52X)
    C      WRITE (L01, 74) PFILES(1:12)
    C      74 FORMAT ( / 5X, 'OUTPUT WILL BE FILED ON ', A12)
    C      GO TO 40
    C      CANNOT OPEN THIS FILE EITHER.
    C      CONTINUE
    C      WRITE (L01, 78) LUFILF
    C      78 FORMAT ( / 5X, 'CANNOT OPEN THIS FILE EITHER. CHECK YOUR FILES ',
    C      1 'FOR THE ABOVE NAME / 5X, 'AND MAKE SURE LOGICAL UNIT', 14,
    C      2 ' IS CLOSED.' / )
    C      IERR = 1
    C      GO TO 210
    C      80 CONTINUE
    C      L02 = LUPRINT

DOUBLE PRECISION PIVOT
COMMON /QPRC8D/ PIVOT

CHARACTER*64 TITLES, PFILES
COMMON /QPRC8C/ TITLES, PFILES

DATA NCALL / 0 /

NCALL = 0

OUTPUT UNITS FOR LINE PRINTER (TERMINAL) AND OUTPUT FILE
DATA LUPRINT, LUFILF / 6, 92 /

INITIALIZATIONS
L01 = LUPRINT
IERR = 0
LTCNT = 0
TZERO = TOL(1)
TPIV = TOL(2)
NCALL = NCALL + 1
DO 10 I = 1, 14
  INPUTS(I) = INPUT(I)
10 CONTINUE
  TITLES =
  *
  L1 = LEN(TITLE)
  L1 = 30
  TITLES(1:L1) = TITLE(1:L1)

INTERNAL TABLEAU IS (MO+NO)*(MO+NO+1) IF ALL INEQUALITIES AND
(MO+NO+1)*(MO+NO+2) IF ANY EQUALITIES SINCE 1 MORE ROW IS ADDED.
IF (MO .LE. 0) GO TO 22
DO 20 I = 1, MO
  IF (KT(I) .EQ. 0) GO TO 24
20 CONTINUE
22 CONTINUE
  MORE = 0
  GO TO 26
24 CONTINUE
  MORE = 1
26 CONTINUE
  P = MO + NO + MORE
  P1 = P + 1
  PSQ = P + 2

SET OUTPUT UNITS AND OPEN PRINT FILE IF REQUIRED.
IF (JOUT .NE. 1 .AND. JOUT .NE. 2) GO TO 80
PFILES =
  *

```

63

```

C
C
C IMPLICIT REAL (A-H,O-Z)
C DIMENSION KT(1)

C
C INTEGER ROW,COL,INDX,P,P1,PSQ
C COMMON /QPCB1/ INPUTS(14), IOUTS(2), ROW,COL,INDX,MORE,P,P1,PSQ,
C 1 NCALL,LENREQ,I01,I02,I03
C EQUIVALENCE (INPUTS(1),ML), (INPUTS(2),NL), (INPUTS(3),MO),
C 1 (INPUTS(4),NO), (INPUTS(5),MINMAX), (INPUTS(6),LENMS),
C 2 (INPUTS(7),MAXIT), (INPUTS(8),KOB), (INPUTS(9),JIT),
C 3 (INPUTS(10),JDATA), (INPUTS(11),JPivot), (INPUTS(12),JSOL),
C 4 (INPUTS(13),JOUT), (INPUTS(14),JWIDTH),
C 5 (IOUTS(1),IERR), (IOUTS(2),ITCNT)
C
C COMMON /QPCB2/ TOLS(2)
C EQU VALENCE (TOLS(1),TZERO), (TOLS(2),TPIV)
C
C INTEGER ZERONE(4),ZERO12(3)
C LOGICAL ERROR(10)
C DATA ZERONE / 5,8,9,11 /
C DATA ZERO12 / 10,12,13 /
C
C ITITLE = 0
C
C DEFAULT VALUES
C IF (MAXIT .LE. 0) MAXIT = 1000
C IF (TZERO .LE. 0.0) TZERO = 1.0E-7
C IF (TPIV .LE. 0.0) TPIV = 1.0E-6
C IF (JWIDTH .LT. 72) JWIDTH = 72
C IF (JWIDTH .GT. 132) JWIDTH = 132
C
C ITITLE = 0
C
C CHECK KT
C DO 10 I = 1, MO
C 10 CONTINUE
C 20 CONTINUE
C WRITE (I01,22) NCALL
C 22 FORMAT (///5X,'QUADPR CALL',I4)
C 24 WRITE (I01,24) (KT(I),I=1,MO)
C 24 FORMAT (/1X,'*** ERROR, ILLEGAL VALUES IN KT. KT CONTAINS'
C 1 / (5112) )
C IERR = 5
C ITITLE = 1
C
C CHECK INPUT
C 30 CONTINUE
C ERROR(1) = (MO .LT. 0 .OR. MO .GT. ML)
C ERROR(2) = (NO .LT. 1 .OR. NO .GT. NL)

```

```

103 = LUPRINT
90 CONTINUE
C
C PARTITION WORKSPACE ARRAY.
C L1 = 1 + 2*(PSQ+P)
C L2 = L1 + 2*P1
C L3 = L2 + 2*P1
C L4 = L3 + P1
C LENREQ = L4 + P1 - 1
C
C CHECK INPUT DATA PARAMETERS FOR CONSISTENCY
C CALL QDRVER (KT)
C
C IF (IERR .EQ. 5) GO TO 200
C QUIT IF BAD VALUES (IERR = 5)
C
C OUTPUT PROBLEM PARAMETERS AND PROBLEM DATA IF REQUESTED.
C IF (JIT .NE. 0) CALL QRPRT1
C IF (JDATA .NE. 0) CALL QRPRT2 (A,ML,KT,RHS,COST,QUAD,NL)
C
C OUTPUT HEADER FOR INTERMEDIATE OUTPUT, IF ANY EXPECTED
C IF (JPivot .NE. 0) CALL QRPRT3
C
C CONSTRUCT INITIAL TABLEAU FOR QUADRATIC PROGRAMMING.
C CALL QDPREP (A,ML,KT,RHS,COST,QUAD,NL,WS,P)
C
C PERFORM PRINCIPAL PIVOTING TO FINAL TABLEAU.
C CALL QDCOMP (WS,WS(L1),WS(L2),WS(L3),WS(L4))
C
C GET SOLUTION AND OBJECTIVE VALUE IF REQUESTED.
C CALL QDSOLN (KT,COST,QUAD,NL,WS,P,WS(L4),X,RC,DUAL,SLK,
C 1 WS(L1),OBJ)
C
C OUTPUT SOLUTION REPORT IF REQUESTED.
C IF (JSOL .NE. 0) CALL QRPRT6 (X,RC,DUAL,SLK,OBJ)
C
C SET EXIT VALUES, IOUT.
C 200 CONTINUE
C IOUT(1) = IERR
C IOUT(2) = ITCNT
C
C CLOSE PRINT FILE IF REQUIRED.
C IF (JOUT .GT. 0) CLOSE (LUFIL,STATUS='KEEP')
C
C RETURN
C
C *****
C END
C
C SUBROUTINE QDRVER (KT)
C
C CHECK INPUT DATA FOR QUADPR

```



```

180      B(K,L) = A(I,L)
      CONTINUE
      B(K,P1) = - RHS(I)
      GO TO 210
C
190      .LE. CONSTRAINT
      CONTINUE
      DO 200 L = 1, NO
      B(K,L) = - A(I,L)
      CONTINUE
      B(K,P1) = RHS(I)
C
210      CONTINUE
C
      FILL {2}.
      DO 230 L = 1, NO
      DO 220 K = N1, P
      B(L,K) = - B(K,L)
      CONTINUE
      DO 220 K = N1, P
      B(L,K) = - B(K,L)
      CONTINUE
C
220      CONTINUE
      B(L,K) = - B(K,L)
      CONTINUE
C
230      CONTINUE
C
      FINISHED
      CONTINUE
      RETURN
      ENI
*****
SUBROUTINE QDCOMP (B,TEMP,U,JZ,JW)
C
C   COMPUTE THE FINAL TABLEAU FOR QUADRATIC PROGRAMMING GIVEN
C   THE INITIAL TABLEAU USING THE METHOD OF PRINCIPAL PIVOTING.
C
      IMP..ICIT REAL (A-H,O-Z)
      DOUBLE PRECISION B,TEMP,U
      INTEGER JZ,JW
      DIMENSION B(1),TEMP(1),U(1),JZ(1),JW(1)
C
      INTEGER ROW,COL,INDX,P,P1,PSQ
      COMMON /QPCB1/ INPUTS(14),TOUTS(2),ROW,COL,INDX,MORE,P,P1,PSQ,
      I,CALL,LENREQ,IOT,IOZ,IO3
      EQUIVALENCE (INPUTS(1),ML), (INPUTS(2),ML), (INPUTS(3),MO),
      (INPUTS(4),MO), (INPUTS(5),MINMAX), (INPUTS(6),LENWS),
      (INPUTS(7),MAXIT), (INPUTS(8),KOBJ), (INPUTS(9),JIT),
      (INPUTS(10),JDATA), (INPUTS(11),JPivot), (INPUTS(12),JSOL),
      (INPUTS(13),JOUT), (INPUTS(14),JWIDTH),
      (TOUTS(1),IERR), (TOUTS(2),ITCNT)
C
      COMMON /QPCBR/ TOLS(2)
      EQUIVALENCE (TOLS(1),ZERO), (TOLS(2),TPIV)
C
      DOUBLE PRECISION PIVOT
      COMMON /QPCBD/ PIVOT

```

```

      L1 = L + 1
      DO 60 K = L1, NO
      B(K,L) = - QUAD(K,L) - QUAD(L,K)
      CONTINUE
      GO TO 70
C
70      CONTINUE
      FILL UPPER TRIANGLE OF {1}.
      DO 80 L = 2, NO
      L1 = L - 1
      DO 90 K = 1, L1
      B(K,L) = B(L,K)
      CONTINUE
      GO TO 100
C
90      CONTINUE
      CHECK FOR CASE OF NO CONSTRAINTS.
      IF (NO.EQ.0) GO TO 240
C
100      CONTINUE
      FILL {5}.
      N1 = NO + 1
      DO 130 L = N1, P
      DO 120 K = N1, P
      B(K,L) = 0.0
      CONTINUE
      GO TO 140
C
120      CONTINUE
      B(K,L) = 0.0
      CONTINUE
      GO TO 150
C
130      CONTINUE
      B(P,P1) = 0.0
      CONTINUE
      FILL {4} AND {6}.
      DO 150 CONTINUE
      DO 210 I = 1, NO
      K = NO + 1
      IF (KT(I).LT.0) GO TO 170
      IF (KT(I).GT.0) GO TO 190
      .EQ. CONSTRAINT
      DO 160 L = 1, NO
      B(K,L) = A(I,L)
      B(P,L) = B(P,L) - B(K,L)
      CONTINUE
      B(K,P1) = - RHS(I)
      B(P,P1) = B(P,P1) - B(K,P1)
      GO TO 210
      .GE. CONSTRAINT
      CONTINUE
      GO TO 170
C
170      CONTINUE
      DO 180 L = 1, NO

```

```

C***** IF (B(IJ) .LT. (-TZERO)) GO TO 90
C***** IF ALL COI. ELEMENTS ARE NONNEGATIVE, THEN THE DRIVING VARIABLE
C***** IS UNBLOCKED AND THERE IS NO SOLUTION.
C***** IF (B(KJ) .GE. (-TPIV)) GO TO 90
C***** Z = -B(IJ) / B(KJ)
C***** IF (Z .GE. THMP) GO TO 90
C***** THMP = Z
C***** ROW = J
C***** 90 CONTINUE
C***** IF (ROW .EQ. 0) GO TO 420
C***** COL = 11
C***** GO TO 150
C***** CONJUGATE Z = -Q(R)/B(R,R). THE INCREASE IN THE DRIVING VARIABLE.
C***** 180 IJ = PSQ + 1
C***** KJ = P(I-1) + 1
C***** Z = -B(IJ) / B(KJ)
C***** COL = I
C***** 110 BS = 0.
C***** ROW = 0.
C***** IF (NEW BLOCKING VARIABLE, BASED ON WHICH BASIC VARIABLE IS
C***** MOST AFFECTED BY THE CHANGE IN THE DRIVING VARIABLE.
C***** IJ = (COL-1)*P
C***** KJ = PSQ
C***** DO 140 J = 1, P
C***** IJ = IJ + 1
C***** KJ = KJ + 1
C***** EST = B(IJ)*Z + B(KJ)
C***** IF (B(KJ) .LT. (-TZERO)) .OR. BST .GE. (-TPIV)) GO TO 140
C***** IF (DABS(B(KJ)) .LT. TZERO .AND. B(IJ) .LT. (-TZERO)) GO TO 120
C***** IF (B(KJ)) 130, 140, 130
C***** ROW = J
C***** GO TO 150
C***** 130 BST = -BST / B(KJ)
C***** IF (BST .LE. BS) GO TO 140
C***** BS = BST
C***** ROW = J
C***** 140 CONTINUE
C***** IF (ROW .EQ. 0) GO TO 210
C***** NONPRINCIPAL PIVOT ON B(S,R).
C***** AT THIS POINT THE BLOCKING VARIABLE IS B(NS,I).
C***** 150 I11 = -JH(ROW)
C***** NFLAG = 1
C***** GO TO 500
C***** 160 CONTINUE
C***** IF (TCNT .GE. MAXIT) GO TO 440
C***** ITCN = ITCN + 1
C***** ATTEMPT TO PROGRESSIVELY RELAX THE CONVERGENCE PARAMETER
C***** TPIV = TPIV + TOLS(2)/50.
C***** INDICATE THE NUMBER OF ITERATIONS
C***** WITH (*,101) OPTIMIZATION ITERATION ,ITCNT

```

```

DOUBLE PRECISION Z,BS,BST
CHARACTER*48 MSG(4)

C DATA MSG / 'ALGORITHM ERROR, INDEX VALUE NON-DECREASING.
C 'NO SOLUTION, INVALID QUADRATIC COSTS MATRIX.
C 'NO SOLUTION, INFEASIBLE OR UNBOUNDED PROBLEM.
C 'ERROR, ITERATION LIMIT REACHED.

C C***** INITIALIZATIONS
C ITCN = 0
C JFLAG = 0
C INDY = P1
C C***** BASIC VARIABLES SET NEGATIVE VALUES.
C C***** NONBASIC VARIABLES (JZ) GET POSITIVE VALUES.
C DO 30 I = 1, P
C JZ(I) = -1.
C JZ(1) = 1
C 30 CONTINUE
C 40 ITEMP = 0
C C***** CHECK FOR NONDECREASING INDEX (ALLOW UP TO 5 EQUAL INDEXES FOR
C TOLERANCE PURPOSES).
C IJ = PSQ
C DO 50 I = 1, P
C IJ = IJ + 1
C IF (B(IJ) .LT. (-TZERO)) ITEMP = ITEMP + 1
C 50 CONTINUE
C IF (ITEMP .NE. INDY) ICT = 0
C IF (ITEMP .EQ. INDY) ICT = ICT + 1
C IF (ITEMP .GT. INDY .OR. ICT .EQ. 5) GO TO 400
C C***** IF THE INDEX IS ZERO YOU ARE DONE.
C IF (ITEMP .EQ. 0) GO TO 300
C INDY = ITEMP
C IJ = PSQ
C DO 60 I = 1, P
C IJ = IJ + 1
C IF (B(IJ) .LT. (-TZERO)) GO TO 70
C 60 CONTINUE
C 70 IJ = P*(I-1) + 1
C C***** CHECK FOR NEGATIVE DIAGONAL ELEMENT (NOT ALLOWED IF POS. SEMIDEF.)
C IF (B(IJ) .LT. (-TPIV)) GO TO 410
C IF (B(IJ) .GT. PIV) GO TO 100
C C***** IF POTENTIAL PIVOT ELEMENT IS NEAR ZERO, USE CODE BELOW
C C***** TO DETERMINE THE BLOCKING VARIABLE.
C I1 = 1
C 80 ROW = 0
C THMP = 1000000.
C IJ = PSQ
C KJ = P*(I1-1)
C DO 90 J = 1, P
C IJ = IJ + 1
C KJ = KJ + 1

```



67

```

C*****EXCHANGE COLUMNS
260 LLL = 0
DO 280 J = 1, P
  K = ABS(JZ(J))
  IF (K.EQ. J) GO TO 280
  IJ = P*(K-1)
  KJ = P*(J-1)
  DO 270 I = 1, P
    IJ = IJ + 1
    KJ = KJ + 1
    Z = B(IJ)
    B(IJ) = B(KJ)
    B(KJ) = Z
  270 CONTINUE
  II = JZ(J)
  JZ(J) = JZ(K)
  JZ(K) = II
  LLL = LLL + 1
280 CONTINUE
IF (LLL.GE. 2) GO TO 260
290 ROW = LL
ITCNT = ITCNT + 1
C*****ATTEMPT TO PROGRESSIVELY RELAX THE CONVERGENCE PARAMETER
C
C      TEIV = TPIV + TOLS(2)/50.
C*****INDICATE THE NUMBER OF ITERATIONS
WRITE(*,101)'OPTIMIZATION ITERATION',ITCNT
C
C      IF (JPIVOT.EQ. 1) CALL QRPRT5
C      IF (INDX.GT. 1) GO TO 40
C*****NORMAL TERMINATION
300 IERR = 1
GO TO 600
C
C*****ERROR TERMINATION
400 K = 1
IERR = 6
GO TO 460
410 K = 2
IERR = 4
GO TO 460
420 K = 3
IERR = 2
GO TO 460
440 K = 4
IERR = 3
460 WRITE (101,470) MSG(K),ITCNT
IF (JOUT.NE. 0) WRITE (103,470) MSG(K),ITCNT
470 FORMAT( / 1X,49B.16,' PIVOTS PERFORMED. / )
GO TO 600
C
101 FORMAT('+',A50,15)
JFLAG = 0
C*****PUT TOLERANCE ON B(I,P) AFTER A NONPRINCIPAL PIVOT.
IJ = PSQ + 1
IF (B(IJ).LT. (-TZERO)) GO TO 170
GO TO 40
C*****THE NEW DRIVING VARIABLE IS THE COMPLEMENT OF THE OLD BLOCKING
C*****VARIABLE
170 DO 180 JJ = 1, P
  IF (JZ(JJ).EQ. III) GO TO 190
180 CONTINUE
C*****NO COMPLEMENT VARIABLE (NOT ALLOWED IF POS. SEMIDEF.)
GO TO 410
190 COL = JJ
C*****THE DISTINGUISHED VARIABLE IS STILL THE I-TH ROW.
IJ = P*(COL-1) + 1
IF (B(IJ).GT. TPIV) GO TO 200
II = COL
GO TO 80
200 KJ = PSQ + 1
Z = -B(KJ) / B(IJ)
GO TO 110
C*****PRINCIPAL PIVOT ON B(I,1).
210 LL = ROW
ROW = I
NFLAG = 2
GO TO 500
220 CONTINUE
IF (JFLAG.EQ. 1) GO TO 290
JFLAG = 1
C*****MUST REARRANGE THE ROWS AND COLUMNS FOR COMPLEMENTARITY
C*****EXCHANGE ROWS
230 LLL = 0
DO 250 I = 1, P
  K = IABS(JW(I))
  IF (K.EQ. I) GO TO 250
  IJ = K
  KJ = I
  DO 240 J = 1, P
    Z = B(IJ)
    B(IJ) = B(KJ)
    B(KJ) = Z
  IJ = IJ + P
  KJ = KJ + P
240 CONTINUE
  II = JW(I)
  JW(I) = JW(K)
  JW(K) = II
  LLL = LLL + 1
250 CONTINUE
IF (LLL.GE. 2) GO TO 230

```

```

*****
SUBROUTINE QDSOLN (KT,COST,QUAD,LQ,B,LDB,JW,X,RC,DUAL,SLK,
1 T1MP,OBJ)
C
C RETURN SOLUTION TO QUADRATIC PROGRAMMING PROBLEM FROM FINAL
C TABLUAU AND COMPUTE OBJECTIVE VALUE IF REQUESTED.
C CONVERTING THE PROBLEM TO MINIMIZATION WITH ALL .GE. CONSTRAINTS.
C
IMPLICIT REAL (A-H,O-Z)

DOUBLE PRECISION B,TEMP
DIMENSION KT(1),COST(1),QUAD(LQ,B,LDB,JW,X(1),RC(1),
1 DUAL(1),SLK(1),TEMP(1))

INTEGER ROW,COL,INDEX,MORE,P,P1,PSQ
COMMON /QPCB1/ INPUTS(14),TOUTS(2),ROW,COL,INDEX,MORE,P,P1,PSQ,
1 NCALL,LENREQ,IOI,IO2,IO3
1 EQUIVALENCE (INPUTS(1),M), (INPUTS(2),M), (INPUTS(3),MO),
2 (INPUTS(4),NO), (INPUTS(5),MINMAX), (INPUTS(6),LENGS),
3 (INPUTS(7),MAXIT), (INPUTS(8),KOB), (INPUTS(9),JIT),
4 (INPUTS(10),JDATA), (INPUTS(11),JPIVOT), (INPUTS(12),JSOL),
5 (INPUTS(13),JOUT), (INPUTS(14),JWIDTH),
6 (TOUTS(1),TERR), (TOUTS(2),ITCNT)

COMMON /QPCB2/ TOLS(2)
EQUIVALENCE (TOLS(1),TZERO), (TOLS(2),TPIV)

DOUBLE PRECISION DART, BB

GET X, RC, DUAL, SLK
100 CONTINUE
KART = MO + 1
DO 11 J = 1, NO
X(J) = 0.0
RC(J) = 0.0
110 CONTINUE
DO 12 I = 1, MO
DUAL(I) = 0.0
SL(I) = 0.0
120 CONTINUE
DART = 0.000
DO 17 I = 1, P
BB = B(1,P)
IF (DABS(BB) .LT. TZERO) BB = 0.000
J = JW(1)
IF (J .LT. 0) GO TO 150
IF (J .GT. NO) GO TO 130
X(J) = BB
GO TO 170
130 CONTINUE
*****PIVOT CODE.
C*****QUADPP USES A NORMAL JORDAN PIVOT.
C*****HOWEVER, WE CAN SAVE MEMORY BY USING THE ELEMENTARY MATRIX.
C*****STORE PIVOT ELEMENT
500 CONTINUE
KC = P*(COL-1)
KJ = KC + ROW
PIVOT = B(KJ)
Z = 1.000 / PIVOT
C*****STORE ELEMENTARY-VECTOR AND U-VECTOR.
IJ = ROW
ISAVE = I
DO 510 I = 1, P1
U(I) = B(IJ)
IJ = IJ + P
510 CONTINUE
IJ = KC
DO 520 I = 1, P
IJ = IJ + 1
TEMP(I) = -B(IJ) * Z
520 CONTINUE
TEMP(ROW) = -Z - 1.000
C*****COMPUTE ALL COLUMNS EXCEPT PIVOT COLUMN.
IJ = 0
DO 550 J = 1, P1
IF (J .NE. COL) GO TO 530
IJ = IJ + P
GO TO 550
530 DO 540 I = 1, P
IJ = IJ + 1
B(IJ) = B(IJ) + U(J)*TEMP(I)
540 CONTINUE
550 CONTINUE
C*****COMPUTE PIVOT COLUMN.
IJ = KC
DO 560 I = 1, P
IJ = IJ + 1
B(IJ) = -TEMP(I)
560 CONTINUE
C*****COMPUTE NEW PIVOT ELEMENT.
B(KJ) = Z
C*****KEEP TRACK OF BASIC AND NONBASIC VARIABLES
K = JW(ROW)
JW(ROW) = JZ(COL)
JZ(COL) = K
IF (JPIVOT .EQ. 1) CALL QRPRT4
I = ISAVE
GO TO (160, 220), NFLAG
C*****FINISHED
600 RETURN
END

```

69

```

*****
SUIROUTINE QSET(NUM,MAXWS,ITURN)
      INCLUDE 'COMMON.F'
      REAL QC(NDID,ND), QRHS(NDID),QTOL(2),QRC(ND),Q2X(ND),XOLD(ND),
      & QDUAL(NDID),QSLK(NDID),QWS(1SIZEQ)
      INTEGER QKT(NDID),INPUT(14),QIOUT(2),NUM,MAXWS
      CHARACTER*30 PFILE

      DO 10 I = 1,10
      DO 10 J = 1,NUM
      QC(I+NUM,J) = C(J,I)
      10 CONTINUE

      DO 20 I = 1,NUM
      DO 20 J = 1,NUM
      IF(I.EQ.J) THEN
      QC(I,J) = -1.0
      ELSE
      QC(I,J) = 0.0
      END IF
      20 CONTINUE

      DO 30 I = 1,NUM+ID
      IF(I.LE.NUM) THEN
      QRHS(I) = -BDU(I)
      ELSE
      QRHS(I) = D(I-NUM)
      END IF
      QKT(I) = -1
      30 CONTINUE

      INPUT(1) = ND+ID
      INPUT(2) = ND
      INPUT(3) = NUM+ID
      INPUT(4) = NUM
      INPUT(5) = 0
      INPUT(6) = MAXWS
      INPUT(7) = 0
      INPUT(8) = 0
      INPUT(9) = 0
      INPUT(10) = 0
      INPUT(11) = 0
      INPUT(12) = 0
      INPUT(13) = 0
      INPUT(14) = 0
      QTOL(1) = 0.0
      C E-6 QTOL(2) = 0.000001
      C E-4 QTOL(2) = 0.0001
      *****
*****
      K = J - NO
      IF (K.EQ. KART) GO TO 140
      DUAL(K) = BB
      GO TO 170
      140 CONTINUE
      DART = BB
      GO TO 170
      150 CONTINUE
      J = -J
      IF (J.GT. NO) GO TO 160
      RC(J) = BB
      GO TO 170
      160 CONTINUE
      K = J - NO
      IF (K.EQ. KART) GO TO 170
      SLK(K) = BB
      170 CONTINUE
      C
      C ADJUST DUALS IF ANY EQUALITIES
      IF (MORE.EQ. 0) GO TO 200
      DO 190 I = 1, NO
      IF (KT(I).NE. 0) GO TO 190
      IF (MINMAX.EQ. 1) GO TO 180
      DUAL(I) = DUAL(I) - DART
      GO TO 190
      180 CONTINUE
      190 CONTINUE
      DUAL(I) = DART - DUAL(I)
      C
      C COMPUTE OBJECTIVE VALUE ONLY IF REQUESTED.
      200 CONTINUE
      IF (KOBJ.EQ. 0) GO TO 250
      OBJ = 0.
      DO 220 I = 1, NO
      TEMP(I) = 0.000
      DO 210 J = 1, NO
      TEMP(I) = TEMP(I) + X(J)*QUAD(I,J)
      210 CONTINUE
      220 CONTINUE
      DO 230 J = 1, NO
      OBJ = OBJ + TEMP(J)*X(J)
      230 CONTINUE
      DO 240 J = 1, NO
      OBJ = OBJ + COST(J)*X(J)
      240 CONTINUE
      C
      C FINISHED
      250 CONTINUE
      RETURN
      END
      *****
*****

```

70

```

60      B(1) = TURN*ABS(X(1)-OLDWT(1))
      CONTINUE
      RETURN
      END
C *****
      PFILE = 'TEMP.Q'
      DO 40 I = 1, NUM
        B(I) = 0.0
        D2X(I) = 0.0
        XOLD(I) = 0.0
      CONTINUE
      DXLAST = 1.0
      CO = 0.1E-3
      C1 = 0.90
      C2 = 1.0
      TURN = TURN * 0.001
      GO TO 501
500      WRITE(*,160)ITURN
150      FORMAT(1X,I60,14)
      CALL QUAD1 (QC,OKT,QRHS,B,A,INPUT,QTOL,TITLE,
      & PFILE,OB,X,QRC,QDUAL,QSLK,QIOUT,QNS)
501      DXTOTAL = 0.0
      DO 45 I = 1, NUM
        IF (ITURN.EQ. 0) THEN
          X(I) = OLDWT(I)
        ELSE
          X(I) = XOLD(I) + C1*(X(I)-XOLD(I))
        END IF
      CONTINUE
45
      DO 50 I = 1, NUM
        A(I,1) = A(I,1) - D2X(I)
        DX = X(I) - OLDWT(I)
        DXTOTAL = DXTOTAL + ABS(DX)
        D2X(I) = 0.5*TURN*(1.0/SQRT(DX*DX+CO)-DX*DX/(DX*DX+CO)**1.5)
        A(I,1) = A(I,1) + D2X(I)
        B(I) = TURN*DX/SQRT(DX*DX + CO) - 2.0*X(I)*D2X(I)
        XOLD(I) = X(I)
      CONTINUE
50
      WRITE(*, '(1X,A12,F9.3)') 'TURN OVER = ', DXTOTAL*50.
      IF (ABS(DXTOTAL-DXLAST).GT. 0.01 .OR. ITURN .LT. 1) THEN
        ITURN = ITURN + 1
        DXLAST = DXTOTAL
        GO TO 500
      END IF
      TURN = 0.0
      DO 60 I = 1, NUM
        A(I,1) = A(I,1) - D2X(I)

```

```

*****
C *** PARAMETERS ARE USED TO DIMENSION THE COMMON ARRAYS      *
C *** IN EACH SUBROUTINE.                                       *
C ***                                                            *

C *** ND = TOTAL NUMBER OF STOCK WEIGHTS BEING COMPUTED.      *
C *** ID = THE MAXIMUM NUMBER OF CONSTRAINT EQUATIONS.        *
C *** N2D = THE DIMENSION ON WORKING AREAS ( 2 * ND )         *
C *** KEQ = NUMBER OF CONSTRAINTS WHICH WILL BE EQUALITIES    *

C *** NRT = NUMBER OF RETURNS                                  *
C *** NST = USED FOR STATISTICS                                *

C *** ISIZEQ = 2 * ( ND*2 + ID )**2 + 8*(ND*2+ID) +6          *
C *** NDID = ND + ID                                           *

C ***** FOR CASH RUNS (AT LEAST 1 EQ CONST. )               *
C *** ISIZEQ = 2 * ( ND*2 + ID )**2 + 12*(ND*2+ID) +16        *
C *** NDID = ND + ID                                           *

C *****
C LAST UPDATED: November 19, 1988
C IMPLICIT REAL (A-H,O-Z)
C IMPLICIT INTEGER (I-N)
C PARAMETER (KEQ = 0)

C 280
C PARAMETER (ND = 280, ID = 6 , N2D = 560)
C PARAMETER (NDID = 286, ISIZEQ = 645246)
C 250
C PARAMETER (ND = 250, ID = 6 , N2D = 500)
C PARAMETER (NDID = 256, ISIZEQ = 516126)
C 100
C PARAMETER (ND = 100, ID = 6 , N2D = 200)
C PARAMETER (NDID = 106, ISIZEQ = 86526)

C PARAMETER (NRT = 48, NST = 3)
C PARAMETER (LENGTH = 40)
C CHARACTER*(LENGTH) FILEOUT,TITLE,NAME(ND),PORTRETS,GARB,ANSWER,
& OLD,LIANAME,LIABIN,STOCKIN,BONDIN,IDC(ND),SIC(99)

COMMON /ARRY/ A(ND,ND),C(ND,ID)
COMMON /VECT/ B(ND),D(ID),BDL(ND),BDU(ND),X(ND),OLDWT(ND)
COMMON /RETS/ RET(NRT,ND),ANIM(NRT),PRET(NRT),ISIC(ND),AVE(ND)
COMMON /TRAC/ COV(NST,NST),STAT(NRT,NST),COREL(NRT,NST)
COMMON /TRA2/ PRICE(ND),VOL(ND),CUM(ND),VAR(3),LIANAME,TITLE
COMMON /RTNIN/FILEOUT,LIABIN,STOCKIN,BONDIN,IDC,NAME,SIC
COMMON /DAT/ NMSAVE,NM,NSIMS,NSTOCKS,NBONDS,NSTAT,NRETS,IDUMMY,
& IHOLD
& COMMON /MAX/ STOCKMIN,YIMAX,SMAX,SCALE,BRET,XFACTOR,XBUPPER,
& XBLOWER,BULLET,TARGET,TURN,BULL1,TARG1,TURN1,PORTVAL
C END OF COMMON.F

```

72

APPENDIX II

LIABILITY RETURN PROGRAM

To create a return series for a liability stream from yields.

(Appendix II pp. 1-2)

73

```

READ(11,1500)GARB
READ(11,1500)GARB
READ(11,*)IYLD1
READ(11,*)IYLD2
DO 333 I=IYLD1,IYLD2
  READ(11,*)YTM(1)
CONTINUE
CLOSE(11)

OPEN (10,FILE=RETOUT,STATUS='NEW')

IF(OLDINT.EQ.1)THEN
  OPEN(8,FILE=OLDFILE)
  READ(8,1500)TITLE
  READ(8,1500)GARB
  READ(8,1500)GARB
  READ(8,1500)GARB
  READ(8,1500)GARB
  READ(8,*)IRET1
  READ(8,*)IRET2
  IF(IRET2.GE.IYLD2)THEN
    WRITE(*,*)'CAN T UPDATE RETURN FILE CHECK YIELDS FILE'
    STOP
  ENDIF
  WRITE(10,1500)TITLE
  WRITE(10,1500)GARB
  WRITE(10,1500)GARB
  WRITE(10,1500)GARB
  WRITE(10,1500)GARB
  WRITE(10,*)IRET1
  WRITE(10,*)IYLD2

DO 500 I=IRET1,IRET2
  READ(8,*)RS,PV1,PV2,MM,DUR
  WRITE(10,200)RS,PV1,PV2,MM,DUR
CONTINUE
CLOSE(8)

ELSE
  IRET1=IYLD1+1
  WRITE(*,*)'ENTER TITLE FOR RETURN FILE'
  READ(1,1500)TITLE
  WRITE(10,1500)TITLE
  WRITE(10,1500)GARB
  WRITE(10,1500)GARB
  WRITE(10,1500)GARB
  WRITE(10,1500)GARB
  WRITE(10,1500)GARB
  WRITE(10,*)IRET1
  WRITE(10,*)IYLD2
ENDIF

SLARGE
SDEBUG
C
C      LIABILITY RETURN PROGRAM
C      AUTHOR JOE DADA III
C      LAST UPDATE 3-21-88
C      COPYWRITE 1988 NATIONAL INVESTMENT SERVICES OF AMERICA INC.

PROGRAM SPANRATE

SINCLUDE: 'RATE.FOR'

C      CALL SUBS TO GET INPUT DATA
C
C      ITYPE=1,ABO;-2,PBO

C      THIS SECTION TO BE REMOVED AND BY SUB INTRO
C      LIAIN='TEMP.LIA'
C      YIELDS='NYTM.PRN'
C      OLDFILE='TEMP.LIR'
C      RETOUT='TEMP2.LIR'
C      OLDOUT=1
C
C      WRITE(*,*)' DOES OLD OUTPUT FILE EXIST ? Y/N'
C      READ(*,1500)GARB
C      IF(GARB.EQ.'Y'.OR.GARB.EQ.'y')THEN
C        OLDOUT=1
C        WRITE(*,*)' ENTER NAME OF OLD RETURN FILE'
C        READ(*,1500)OLDFILE
C      ELSE
C        OLDOUT=0
C      ENDIF
C      WRITE(*,*)' ENTER LIABILITY STREAM FILENAME'
C      READ(*,1500)LIAIN
C      WRITE(*,*)' ENTER NEW OUTPUT FILENAME'
C      READ(*,1500)RETOUT
C      WRITE(*,*)' RETOUT'
C      WRITE(*,*)' ENTER YIELD TO MATURITY FILENAME'
C      READ(*,1500)YIELDS

OPEN (11,FILE=YIELDS)
READ(11,1500)GARB
READ(11,1500)GARB
READ(11,1500)GARB
READ(11,1500)GARB

```

```

C
C
IF OLDFILE EXIST CHECK LAST RETURN IN IT
VERSUS NEW CALC TO CHECK PARAMETERS

OPEN (9, FILE-LIAIN)
DO 448 I=1,1000
  READ(9,*,END = 449)XLIA(I)
CONTINUE
CLOSE(9)

448
449
NYEARS=I-1
NMONTHS=NYEARS*12
IF(NYEARS.GT.90)THEN
  WRITE(*,*)' ERROR LIA STREAM HAS TOO MANY YRS 90 IS MAX'
  STOP
ENDIF

C
C
      CALCULATE MONTHLY YIELDS AND SLIDE
      OLD YIELD BACK TO BEG OF MONTH

DO 222 I=1,NYEARS
  XLIA(I)=XLIA(I)/12
CONTINUE

222
IF(OLDOUT.EQ.1)THEN
  ISTART=IRET2-1
ELSE
  ISTART=IYLDI
ENDIF
ICHECK=ISTART+1
DO 100 I=ISTART,IYLD2
  WRITE(*,*)I,I,'YLD',YTM(I)
  XLIA(I)
+
  FR=FR
  FR=(1+YTM(I))**(0.083333)-1

C
C
      SKIP TO END OF LOOP FIRST TIME THROUGH
      IF(FR1.EQ.0)GO TO 100
      PROD IS DENOMINATOR TO CONV FUTURE TO PV

C
C
  PV1=0.0
  PV2=0.0
  PROD1=1.0
  PROD2=1.0
  DUR=0
  DO 300 J=1,NYEARS
    DO 300 I=1,12
      PROD1=PROD1*(1+FR1)
      PV1=PV1+XLIA(J)/PROD1
      DUR=DUR+(J+1)/12*XLIA(J)/PROD1
      IF(J.EQ.1.AND.L.EQ.1)GO TO 400
      PROD2=PROD2*(1+FR)
      PV2=PV2+XLIA(J)/PROD2
    CONTINUE
  RR=PV2/PV1-1.0
  DUR=DUR/PV1

400
300

```



75

APPENDIX III

CORRELATION PORTFOLIO PROGRAM

To create an optimal index correlation portfolio with securities.

(Appendix III pp. 1-30)

!

77

```

DO 20 J = 1,18
  READ(13, '(A)', END = 99) GARB
  CONTINUE
20  CONTINUE
30  REAL(13,101, END = 99) TITLE
    REAL(13,101, END = 99) FILEOUT
    REAL(13,101, END = 99) LTABIN
    REAL(13,101, END = 99) STOCKIN
    READ(13,101, END = 99) BONDIN
    READ(13,101, END = 99) BONDIN
    READ(13,101, END = 99) NSAVE
    READ(13,101, END = 99) NSIMS
    READ(13,101, END = 99) NSTOCKS
    READ(13,101, END = 99) NBONDS
    READ(13,101, END = 99) IDUMMY
    READ(13,101, END = 99) STOCKMIN
    READ(13,101, END = 99) YIMAX
    READ(13,101, END = 99) SMAX
    READ(13,101, END = 99) NSTAT
    READ(13,101, END = 99) NRETS
    READ(13,101, END = 99) BULLET
    READ(13,101, END = 99) TARGET
    READ(13,101, END = 99) TURN
    101  FORMAT (A20)
    ISKIP = ISKIP+1
    CLOSE(13)
    RETURN
C ***** STOP DONE WITH MULTI RUN *****
99  CLOSE(13)
    WRITE(*,*) 'DONE WITH MULTIPLE RUN *****'
    STOP
    END
*****
***** END OF SUBROUTINE MULTRUN *****
*****
***** SUBROUTINE HOLDING (NUM) *****
*****
***** INCLUDE 'CONSUC.F' *****
*****
***** DIMENSION VAL(ND) *****
***** CHARACTER*9 SYMBL, IDCYN(ND) *****
***** CHARACTER*12 HOLD *****
***** LOGICAL HOLDEX *****
*****
C ***** HOLD IS READ FROM FIRST SCREEN RESPONSE: OLD OR HOLDINGS RUN **
  IF (I HOLD .EQ. 0) GO TO 999
11  WRITE(*,*) 'INPUT PORTFOLIO #, AND NAME OF HOLDINGS FILE'

```

```

MAXWS = 2*(2*NUM+ID)*(2*NUM+ID)+8*(2*NUM+ID)+6
ITURN = 0
CALL QSET ( NUM, MAXWS, ITURN )
***** IF TURNOVER IS CONSIDERED, CALL TURNOV
C
  IF (TURN.GT.0) THEN
    ITURN = 1
    CALL QSET (NUM, MAXWS, ITURN)
    END IF
*****
C ***** CALL TO PORTFOLIO OUTPUT *****
  WRITE(*,*)
  WRITE(*,101) 'CALLING PORT *****'
  CALL PORT (NUM)
  IF (IDUMMY .EQ. 1) GO TO 100
C ***** CALL TO STATISTICAL TRACKING *****
  WRITE(*,101) 'CALLING TRACKER *****'
  CALL TRACKER (NUM, IREAD)
C ***** CALL TO TURNOVER ROUTINE *****
  WRITE(*,101) 'CALLING BAL *****'
  CALL BAL (NUM)
C *****
100  WRITE(*,101) 'SPANNING SIMULATION COMPLETE'
    IF (ICOUNT.EQV. .TRUE.) THEN
      WRITE(*,102) 'DONE WITH RUN NUMBER ', ISKIP
      GO TO 5
    END IF
101  FORMAT(' ', A50)
102  FORMAT(' ', A50, I7)
C
C ***** END OF MAIN PROGRAM SPANI *****
C ***** SUBROUTINE MULTRUN (ISKIP) *****
*****
***** INCLUDE 'CONSUC.F' *****
***** CHARACTER*30 XTITL, XFLOUT, XLBIN, XSTKIN, XBNDIN *****
***** OPEN (13, FILE = 'MULTIN', STATUS = 'OLD') *****
***** IF (ISKIP.EQ.0) GO TO 30 *****
***** DO 10 I = 1, ISKIP *****

```

78

```

IDCTEST = 0
NCOUNT = 0
DO 20 I = 2, NIDC
  ITEIT = 0
  DO 10 J = 1, NUM
    IF (IDC(J)(1:4) .EQ. IDC SYN(1)) THEN
      ITEST = 1
      NCOUNT = NCOUNT + 1
      OLDNT(J) = VAL(1) / PORTVAL
    END IF
  CONTINUE
  IF (ITEST .EQ. 0) THEN
    IF (IDCTEST .EQ. 1) THEN
      WRITE(*,*) IDC SYN(1), ': SYMBOL NOT INCLUDED IN RETURN FILE'
    END IF
  CONTINUE
20 CONTINUE

IF (IDCTEST .EQ. 1) THEN
  WRITE(*,*) NIDC - NCOUNT - 1, 'SYMBOLS NOT FOUND..PROGRAM STOPPING'
  STOP
END IF
RETURN

999 OPEN(21, FILE = 'OLD', STATUS = 'OLD')
DO 90 I = 1, NUM
  READ(21, *) OLDNT(I)
  IF (TURN .EQ. 0.0) OLDNT(I) = 0.0
  OLDNT(I) = OLDNT(I) / 100.
90 CONTINUE
CLOSE(21)
RETURN

END
C *****
C UPDATE 8/9/88
C *****
C THIS SUBROUTINE READS IN STOCK, BOND & LIA DATA *****
C *****
C SUBROUTINE COVIN (NUM, IREAD, IFIRST, ILAST)
C *****
C IFIRST IS FIRST MONTH OF DATA NEEDED FOR MATRIX
C ILAST IS LAST MONTH OF SIMULATION
C *****
C INCLUDE 'COMSUC.F'
C *****
C READ IN LIABILITY RETURNS *****
C *****
OPEN (9, FILE = 'LIABIN', STATUS = 'OLD')
READ (9, 111) LIANAME

```

```

READ(*,*) IPORT, HOLD
INQUIRE (FILE = HOLD, EXIST = HOLD EX)
IF (HOLD EX .EQV. .FALSE.) THEN
  WRITE(*,*) 'FILE NOT FOUND', HOLD
  GO TO 11
END IF

OPEN (25, FILE = HOLD, STATUS = 'OLD')
XPORT = REAL(IPORT)
I = 0
READ(25, *, END = 44) SYMBL VALUE
IF (SYMBL .EQ. IDC SYMBL) THEN
  IF (VALUE .EQ. XPORT) THEN
    READ(25, *, END = 44) GARB
    READ(26, *, END = 44) SYMBL VALUE
    IF (SYMBL .EQ. IDC SYMBL) GO TO 44
  I = I + 1
  IDC SYMBL(1) = SYMBL
  VAL(1) = VALUE
  GO TO 33
END IF
READ(25, *, END = 44) GARB
END IF
GO TO 22

CONTINUE
NIDC = I
IF (NIDC .EQ. 0) THEN
  WRITE(*,*) 'PORTFOLIO NUMBER NOT FOUND', IPORT
  GO TO 11
END IF

ICASH = 0
PORTVAL = 0.0
DO 10 I = 1, NIDC
  PORTVAL = PORTVAL + VAL(I)
  IF (IDC SYMBL(1) .EQ. 'CASH') THEN
    ICASH = I
    TEMP = VAL(I)
    VAL(I) = VAL(I)
    VAL(I) = TEMP
    GARB = IDC SYMBL(1)
    IDC SYMBL(1) = IDC SYMBL(1)
    IDC SYMBL(1) = GARB
  END IF
CONTINUE
IF (ICASH .EQ. 0) THEN
  WRITE(*,*) 'NO CASH INCLUDED.....PROGRAM STOPPING'
  STOP
END IF

```

```

791 ILEN = 1  
IDC(1) =  
IF (GARB(ILEN:ILEN).EQ.' ') THEN  
ILEN = ILEN + 1  
IF ((ILEN+3).GT.LENGTH) GOTO 793  
GOTO 791  
ENDIF  
  
792 IDC(1)(1:4) = GARB(ILEN:ILEN+3)  
ILEN = ILEN + 4  
  
IF (GARB(ILEN:ILEN).EQ.' ') THEN  
ILEN = ILEN + 1  
IF ((ILEN+3).GT.LENGTH) GOTO 793  
GOTO 792  
ENDIF  
IDC(1)(6:9) = GARB(ILEN:ILEN+3)  
  
793 CONTINUE  
  
C *****  
TEST FOR ENOUGH RETURNS TO SATISFY NRETS *****  
  
ICHECK = NM-NRETS-ISTART  
IF(ICHECK.LT.O) NRETS = NM - ISTART  
IF(NRETS.LT.NSTAT) ICHECK = IFIRST-ISTART  
  
DO 800 J = 1,ICHECK  
READ (4,*) GARB  
CONTINUE  
  
DO 801 J = 1,NRETS-NSTAT  
READ(4,*)GARB  
CONTINUE  
  
JJ = NRETS-NSTAT  
  
DO 810 K = 1,IREAD  
READ (4,*) RET(K,1)  
  
***** ADJUST OCTOBER 1987 *****  
IF (K+NM-NSTAT-1.EQ.94.) THEN  
IF (K.-E.NSTAT)RET(K,1)=RET(K,1)/2.97  
END IF  
  
J = JJ+K  
CONTINUE  
  
B10 C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
READ (4,111) GARB  
  
C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
WRITE(*,*)LIANAME  
DO 200 I = 1,5  
READ (9,111) GARB  
CONTINUE  
  
200 CONTINUE  
  
READ {9,*} ISTART  
READ {9,*} ISTOP  
  
DO 300 I = 1,IFIRST-ISTART  
READ (9,111) GARB  
CONTINUE  
  
300 CONTINUE  
  
DO 400 I = 1,IREAD  
READ (9,*) ANIM(I)  
CONTINUE  
  
400 CONTINUE  
  
CLOSE(9)  
  
C READ IN STOCK DATA  
IF (NSTOCKS.GT.O) THEN  
OPEN (4,FILE = STOCKIN, STATUS = 'OLD')  
  
DO 600 I = 1,TDUMMY  
READ (4,111) (GARB,L = 1,6)  
READ (4,*) ISTART  
READ (4,*) ISTOP  
DO 610 K = 1,ISTOP-ISTART+1  
READ (4,*) XGARB  
CONTINUE  
FORMAT (A30)  
  
610 CONTINUE  
600 CONTINUE  
111 FORMAT (A30)  
  
DO 700 I = 1,NSTOCKS  
READ (4,111,END = 785) NAME(I)  
  
783 C ***** CUT OFF LEADING BLANKS OF NAME(I)  
ILEN = 1  
GARB =  
IF (NAME(I){ILEN:ILEN}.EQ.' ')THEN  
ILEN = ILEN+1  
GO TO 790  
ELSE  
GARB(1:LENGTH) = NAME(I)(ILEN:LENGTH)  
END IF  
NAME(I) = GARB  
  
790 WRITE (*,112) I,NAME(I)  
FORMAT(*,*,STOCK #,{15,' IS ',A30)  
READ (4,111) GARB  
  
112 C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
READ (4,111) GARB  
  
C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
WRITE(*,*)LIANAME  
DO 200 I = 1,5  
READ (9,111) GARB  
CONTINUE  
  
200 CONTINUE  
  
READ {9,*} ISTART  
READ {9,*} ISTOP  
  
DO 300 I = 1,IFIRST-ISTART  
READ (9,111) GARB  
CONTINUE  
  
300 CONTINUE  
  
DO 400 I = 1,IREAD  
READ (9,*) ANIM(I)  
CONTINUE  
  
400 CONTINUE  
  
CLOSE(9)  
  
C READ IN STOCK DATA  
IF (NSTOCKS.GT.O) THEN  
OPEN (4,FILE = STOCKIN, STATUS = 'OLD')  
  
DO 600 I = 1,TDUMMY  
READ (4,111) (GARB,L = 1,6)  
READ (4,*) ISTART  
READ (4,*) ISTOP  
DO 610 K = 1,ISTOP-ISTART+1  
READ (4,*) XGARB  
CONTINUE  
FORMAT (A30)  
  
610 CONTINUE  
600 CONTINUE  
111 FORMAT (A30)  
  
DO 700 I = 1,NSTOCKS  
READ (4,111,END = 785) NAME(I)  
  
783 C ***** CUT OFF LEADING BLANKS OF NAME(I)  
ILEN = 1  
GARB =  
IF (NAME(I){ILEN:ILEN}.EQ.' ')THEN  
ILEN = ILEN+1  
GO TO 790  
ELSE  
GARB(1:LENGTH) = NAME(I)(ILEN:LENGTH)  
END IF  
NAME(I) = GARB  
  
790 WRITE (*,112) I,NAME(I)  
FORMAT(*,*,STOCK #,{15,' IS ',A30)  
READ (4,111) GARB  
  
112 C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
READ (4,111) GARB  
  
C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
WRITE(*,*)LIANAME  
DO 200 I = 1,5  
READ (9,111) GARB  
CONTINUE  
  
200 CONTINUE  
  
READ {9,*} ISTART  
READ {9,*} ISTOP  
  
DO 300 I = 1,IFIRST-ISTART  
READ (9,111) GARB  
CONTINUE  
  
300 CONTINUE  
  
DO 400 I = 1,IREAD  
READ (9,*) ANIM(I)  
CONTINUE  
  
400 CONTINUE  
  
CLOSE(9)  
  
C READ IN STOCK DATA  
IF (NSTOCKS.GT.O) THEN  
OPEN (4,FILE = STOCKIN, STATUS = 'OLD')  
  
DO 600 I = 1,TDUMMY  
READ (4,111) (GARB,L = 1,6)  
READ (4,*) ISTART  
READ (4,*) ISTOP  
DO 610 K = 1,ISTOP-ISTART+1  
READ (4,*) XGARB  
CONTINUE  
FORMAT (A30)  
  
610 CONTINUE  
600 CONTINUE  
111 FORMAT (A30)  
  
DO 700 I = 1,NSTOCKS  
READ (4,111,END = 785) NAME(I)  
  
783 C ***** CUT OFF LEADING BLANKS OF NAME(I)  
ILEN = 1  
GARB =  
IF (NAME(I){ILEN:ILEN}.EQ.' ')THEN  
ILEN = ILEN+1  
GO TO 790  
ELSE  
GARB(1:LENGTH) = NAME(I)(ILEN:LENGTH)  
END IF  
NAME(I) = GARB  
  
790 WRITE (*,112) I,NAME(I)  
FORMAT(*,*,STOCK #,{15,' IS ',A30)  
READ (4,111) GARB  
  
112 C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
READ (4,111) GARB  
  
C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
WRITE(*,*)LIANAME  
DO 200 I = 1,5  
READ (9,111) GARB  
CONTINUE  
  
200 CONTINUE  
  
READ {9,*} ISTART  
READ {9,*} ISTOP  
  
DO 300 I = 1,IFIRST-ISTART  
READ (9,111) GARB  
CONTINUE  
  
300 CONTINUE  
  
DO 400 I = 1,IREAD  
READ (9,*) ANIM(I)  
CONTINUE  
  
400 CONTINUE  
  
CLOSE(9)  
  
C READ IN STOCK DATA  
IF (NSTOCKS.GT.O) THEN  
OPEN (4,FILE = STOCKIN, STATUS = 'OLD')  
  
DO 600 I = 1,TDUMMY  
READ (4,111) (GARB,L = 1,6)  
READ (4,*) ISTART  
READ (4,*) ISTOP  
DO 610 K = 1,ISTOP-ISTART+1  
READ (4,*) XGARB  
CONTINUE  
FORMAT (A30)  
  
610 CONTINUE  
600 CONTINUE  
111 FORMAT (A30)  
  
DO 700 I = 1,NSTOCKS  
READ (4,111,END = 785) NAME(I)  
  
783 C ***** CUT OFF LEADING BLANKS OF NAME(I)  
ILEN = 1  
GARB =  
IF (NAME(I){ILEN:ILEN}.EQ.' ')THEN  
ILEN = ILEN+1  
GO TO 790  
ELSE  
GARB(1:LENGTH) = NAME(I)(ILEN:LENGTH)  
END IF  
NAME(I) = GARB  
  
790 WRITE (*,112) I,NAME(I)  
FORMAT(*,*,STOCK #,{15,' IS ',A30)  
READ (4,111) GARB  
  
112 C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
READ (4,111) GARB  
  
C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
WRITE(*,*)LIANAME  
DO 200 I = 1,5  
READ (9,111) GARB  
CONTINUE  
  
200 CONTINUE  
  
READ {9,*} ISTART  
READ {9,*} ISTOP  
  
DO 300 I = 1,IFIRST-ISTART  
READ (9,111) GARB  
CONTINUE  
  
300 CONTINUE  
  
DO 400 I = 1,IREAD  
READ (9,*) ANIM(I)  
CONTINUE  
  
400 CONTINUE  
  
CLOSE(9)  
  
C READ IN STOCK DATA  
IF (NSTOCKS.GT.O) THEN  
OPEN (4,FILE = STOCKIN, STATUS = 'OLD')  
  
DO 600 I = 1,TDUMMY  
READ (4,111) (GARB,L = 1,6)  
READ (4,*) ISTART  
READ (4,*) ISTOP  
DO 610 K = 1,ISTOP-ISTART+1  
READ (4,*) XGARB  
CONTINUE  
FORMAT (A30)  
  
610 CONTINUE  
600 CONTINUE  
111 FORMAT (A30)  
  
DO 700 I = 1,NSTOCKS  
READ (4,111,END = 785) NAME(I)  
  
783 C ***** CUT OFF LEADING BLANKS OF NAME(I)  
ILEN = 1  
GARB =  
IF (NAME(I){ILEN:ILEN}.EQ.' ')THEN  
ILEN = ILEN+1  
GO TO 790  
ELSE  
GARB(1:LENGTH) = NAME(I)(ILEN:LENGTH)  
END IF  
NAME(I) = GARB  
  
790 WRITE (*,112) I,NAME(I)  
FORMAT(*,*,STOCK #,{15,' IS ',A30)  
READ (4,111) GARB  
  
112 C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
READ (4,111) GARB  
  
C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
WRITE(*,*)LIANAME  
DO 200 I = 1,5  
READ (9,111) GARB  
CONTINUE  
  
200 CONTINUE  
  
READ {9,*} ISTART  
READ {9,*} ISTOP  
  
DO 300 I = 1,IFIRST-ISTART  
READ (9,111) GARB  
CONTINUE  
  
300 CONTINUE  
  
DO 400 I = 1,IREAD  
READ (9,*) ANIM(I)  
CONTINUE  
  
400 CONTINUE  
  
CLOSE(9)  
  
C READ IN STOCK DATA  
IF (NSTOCKS.GT.O) THEN  
OPEN (4,FILE = STOCKIN, STATUS = 'OLD')  
  
DO 600 I = 1,TDUMMY  
READ (4,111) (GARB,L = 1,6)  
READ (4,*) ISTART  
READ (4,*) ISTOP  
DO 610 K = 1,ISTOP-ISTART+1  
READ (4,*) XGARB  
CONTINUE  
FORMAT (A30)  
  
610 CONTINUE  
600 CONTINUE  
111 FORMAT (A30)  
  
DO 700 I = 1,NSTOCKS  
READ (4,111,END = 785) NAME(I)  
  
783 C ***** CUT OFF LEADING BLANKS OF NAME(I)  
ILEN = 1  
GARB =  
IF (NAME(I){ILEN:ILEN}.EQ.' ')THEN  
ILEN = ILEN+1  
GO TO 790  
ELSE  
GARB(1:LENGTH) = NAME(I)(ILEN:LENGTH)  
END IF  
NAME(I) = GARB  
  
790 WRITE (*,112) I,NAME(I)  
FORMAT(*,*,STOCK #,{15,' IS ',A30)  
READ (4,111) GARB  
  
112 C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
READ (4,111) GARB  
  
C ***** CUT OUT BLANKS FROM GARB TO GET AN IDC AND TICKER  
WRITE(*,*)LIANAME  
DO 200 I = 1,5  
READ (9,111) GARB  
CONTINUE  
  
200 CONTINUE  
  
READ {9,*} ISTART  
READ {9,*} ISTOP  
  
DO 300 I = 1,IFIRST-ISTART  
READ (9,111) GARB  
CONTINUE  
  
300 CONTINUE  
  
DO 400 I = 1,IREAD  
READ (9,*) ANIM(I)  
CONTINUE  
  
400 CONTINUE  
  
CLOSE(9)  
  
C READ IN STOCK DATA  
IF (NSTOCKS.GT.O) THEN  
OPEN (4,FILE = STOCKIN, STATUS = 'OLD')  
  
DO 600 I = 1,TDUMMY  
READ (4,111) (GARB,L = 1,6
```

80

```

CLOSE(7)
RETURN

C *** IF END OF FILE WAS HIT ON STOCKS - RESET NUM AND CONTINUE ***
785 NSTOCKS = I-1
    NUM = NSTOCKS+NBONDS
    GOTO 1400

END
C ***** END OF SUBROUTINE COVIN *****
C ***** MATRIX CALCULATION SUBROUTINE *****
C ***** SUBROUTINE MATRIX (NUM, IREAD) *****
C ***** INCLUDE 'COMSUC.F.' *****
C ***** IREAD = NSTOCKS+NBONDS *****
C ***** NUM = NSTOCKS+NBONDS *****
    UANIM = 0.0
    DO 198 I = 1, NUM
        AVE(I) = 0.0
        DO 199 J = 1, NSTAT
            AVE(I) = AVE(I) + RET(J, I) / DBLE(NSTAT)
        CONTINUE
    CONTINUE
    UANIM = 0.
    DO 120 I = 1, NSTAT
        UANIM = UANIM + ANIM(I) * TARGET / DBLE(NSTAT)
    CONTINUE
    XXX = DBLE(NSTAT-1)
    DO 130 I = 1, NUM
        B(I) = 0.0
        DO 131 J = 3, NSTAT
            TEMPI = (1+RET(J, I)) * (1+RET(J-1, I)) * (1+RET(J-2, I)) - 1.0
            TEMPJ = (1+ANIM(J)) * (1+ANIM(J-1)) * (1+ANIM(J-2)) - 1.0
            B(I) = B(I) + (TEMPI - AVE(I) * 3.) * (TEMPI - TARGET - UANIM * 3.) / XXX
        CONTINUE
    CONTINUE
C NOW CALCULATES COVAR AND SIGMA PURE
CCCC REMOVE COMMENTS DOWN TO CCCCC LINE TO USE SIGMA DIFF
C
C DO 233 I = 1, NUM
C     OLDWT(I) = OLDWT(I) / 100.
C     BRET = BRET + OLDWT(I) * OLDWT(I)
C     DO 244 J = 1, NSTAT
C         RET(J, I) = RET(J, I) - ANIM(J) * TARGET
C ***** TEST ICHCK FOR ENOUGH RETURNS FOR BRET *****
C ***** ICHCK = NM-NRETS-ISTART *****
C ***** IF(ICHCK.LT.0) THEN *****
C ***** WRITE(*,*) BOND, 1, DOES NOT HAVE ENOUGH RETURNS FOR NRETS *****
C ***** STOP *****
C ***** END IF *****
C ***** IF(NRETS.LT.NSTAT) ICHCK = IFIRST-ISTART *****
    DO 1000 J = 1, ICHCK
        READ(7, 111) GARB
    CONTINUE
    DO 1001 J = 1, NRETS-NSTAT
        READ(7, *) GARB
    CONTINUE
    JJ = NRETS-NSTAT
    DO 1010 K = 1, IREAD
        READ(7, *) RET(K, I)
        J = JJ + K
    CONTINUE
    DO 1300 J = 1, ISTOP-ILAST
        READ(7, 111) GARB
    CONTINUE
    DO 900 CONTINUE
    ENDIF

```

```

387 CONTINUE
      RETURN
      END
      END OF SUBROUTINE MATRIX
C*****
C***** SUBROUTINE BOUNDS *****
C***** SUBROUTINE BOUNDS (NUM,ICOL,MM) *****
C***** SUBROUTINE BOUNDS (NUM,ICOL,MM) *****
C*****
C      INCLUDE 'CONSUC.F'

```

82

```

      RETURN
      END IF
C ***** PRINT SPANNING SIMULATION RESULTS *****
      OPEN(10,FILE=FILEOUT,STATUS='UNKNOWN')
      REWIND(10)
      WRITE(10,*) 'NATIONAL INVESTMENT SERVICES SPANNING TECHNOLOGY'
      WRITE(10,*) 'SIMULATION RESULTS'
      WRITE(10,*) 'TITLE'
      WRITE(10,*) 'LIABILITY STREAM USED',LIANAME
      WRITE(10,*) 'START END SPANNING'
      WRITE(10,*) 'NUMBER WEIGHT SENSITIVITY SIC # IDC'
      WRITE(10,*) 'SECURITY NAME'
      WRITE(10,*)
C ***** CALCULATE THE PORTFOLIO RETURNS *****
C ***** FIRST CALCULATE RETURNS FOR NSTAT PERIOD *****
      DO 800 I = 1,NSTAT
      PRET(I) = 0.0
      DO 900 J = 1,NUM
      PRET(I) = PRET(I) + RET(I,J)*X(J)
      CONTINUE
      DO 801 I = 1,NSINS
      PRET(I+NSTAT) = 0.0
      DO 901 J = 1,NUM
      PRET(I+NSTAT) = PRET(I+NSTAT) + CUM(J)*RET(I+NSTAT,J)
      CONTINUE
      DO 1001 K = 1,NUM
      CUM(K) = CUM(K) * (1. + RET(I+NSTAT,K)) / (1. + PRET(I+NSTAT))
      CONTINUE
C *** WRITE OUT THE NEW WEIGHTS TO A FILE CALLED 'OLD' *****
      OPEN(11,FILE='OLD',STATUS='OLD')
      REWIND(11)
      DO 140 I = 1,NUM
      WRITE(11,940) CUM(I)*100
      CONTINUE
      FORMAT(1X,F15.8)
      CLOSE(11)
C ***** CALCULATE PARTIALS AND WRITE OUTPUT *****
      XVAR = 0.0
      XCOVAR = 0.0
      XRET = 0.0
      IF (J.EQ.5.AND.ISIC(1).EQ.67) C(1,3) = -1.0
      IF (J.EQ.6.AND.ISIC(1).EQ.67) C(1,3) = -1.0
      CONTINUE
      CONTINUE
      CLOSE (6)
      MMN = 2*NUM+ICOL
      RETURN
      END
      ***** END OF SUBROUTINE BOUNDS *****
C ***** SUBROUTINE PORT (NUM) *****
C ***** INCLUDE 'CONSUC.F' *****
C ***** BEGIN OUTPUT PROCEDURES *****
      UANIM = 0.0
      DO 129 I = 1,NRETS
      UANIM = UANIM + ANIM(I)/DBLE(NRETS)
      CONTINUE
      UANIM = UANIM*TARGET
      XTOT=0.
      IDUMMY = 0
      DO 130 I=1,NUM
      XTOT=XTOT+X(I)
      CUM(I) = X(I)
      IF (X(I).LT. -0.001) THEN
      WRITE(*,*) '***** SPANNING RUN ABORTED *****'
      WRITE(*,*) 'WEIGHT NUMBER',I,'= ',X(I)*100.
      WRITE(*,*) '*****'
      IDUMMY = 1
      RETURN
      END IF
      CONTINUE
      IF (ABS(XTOT-1.0) .GT. 0.005) THEN
      WRITE(*,*) '***** SPANNING RUN ABORTED *****'
      WRITE(*,*) 'TOTAL WEIGHT IN PORTFOLIO = ',XTOT*100.
      WRITE(*,*) '*****'
      IDUMMY = 1
      RETURN
      END IF
      CONTINUE

```



83

```

C
C      DO 1300 I=1,NSTAT
C      RET(I,J)=RET(I,J)+ANIM(I)*TARGET
C      IF(I+NM-NSTAT-.1.EQ.94)RET(I,J)=RET(I,J)+2.97
C 1300 CONTINUE

C ***** SET AVERAGES AND COVARIANCES TO ZERO *****
DO 400 I=1,3
  AVE(I)=0.0
  DO 400 J=1,3
    COV(I,J)=0.0
  400 CONTINUE

C *** SET STAT ARRAY 1 = LIABILITY, 2 = PORTFOLIO, 3 = DIFFERENCES **
DO 250 I=1,IREAD
  STAT(I,1)=ANIM(I)
  STAT(I,2)=PRET(I)
  STAT(I,3)=PRET(I)-ANIM(I)
250 CONTINUE

C ***** CALCULATE AVERAGES *****
NSIM = NSIMS
IF(NSIMS.LT.1) NSIM = 2
DO 260 I=1,NSTAT+1 IREAD
  AVE(I)=AVE(I)+ANIM(I)/DBLE(NSIM)
  AVE(2)=AVE(2)+PRET(I)/DBLE(NSIM)
  AVE(3)=AVE(3)+PRET(I)-ANIM(I)/DBLE(NSIM)
260 CONTINUE

C ***** CALCULATE COVARIANCES *****
DO 950 I=1,3
  VAR(I)=0.0
  DO 950 K=NSTAT+1,IREAD
    VAR(I)=VAR(I)+(STAT(K,I)-AVE(I))**2/DBLE(NSIM-1)
  950 CONTINUE

C ***** CALCULATE THE COVARIANCE MATRIX'S UPPER OFF DIAGONAL ****
DO 960 I=1,3
  DO 960 J=1,3
    DO 828 K=NSTAT+1,IREAD
      COV(I,J)=COV(I,J)+(STAT(K,I)-AVE(I))*(STAT(K,J)-AVE(J))
    CONTINUE
    COV(I,J)=COV(I,J)/DBLE(NSIM-1)
    COV(J,I) = COV(I,J)
  960 CONTINUE

C ***** CALCULATE CORRELATIONS *****

```

```

1:COUNT=0
DO 131 I=1,NUM
  IF(BULLET.NE.0.0)B(I) = B(I)/BULLET
  C(I,1)=0.0
  DO 132 J=1,NUM
    C(I,1)=C(I,1)+X(J)*2.0*A(I,J)
    XVAR=XVAR+X(I)*X(J)*A(I,J)
  132 CONTINUE
  XRET1 = AVE(1) + UANIM
  XRET = XRET + X(I)*XRET1
  XCOVAR = XCOVAR + X(I)*B(I)
131 CONTINUE

DO 150 I = 1, NUM
  C(I,1) = B(I)*XVAR**(-0.5)-0.5*XCOVAR*C(I,1)*XVAR**(-1.5)
  ICOUNT=ICOUNT+1
  WRITE(10,16)ICOUNT,X(I)*100,CUM(I)*100,C(I,1)*100,
  &
  CONTINUE
  FORIAT(IX,14,1X,2F8.3,F10.6,14,2X,A9,2X,A30)
  WRI'E(10,*)

  STD=(XVAR*12)**(.5)*100
  XRET1 = XRET*12.0*100.0
  XCOVAR = XCOVAR*1200

  WRITE(10,649) STD, XRET, XCOVAR
  FORMAT(1X,' MINIMUM STD DEV =',F10.4,' EXPECTED RETURN =',F10.4,
  &,' VARIANCE =',F10.4)
  WRITE(10,*)

C IF(NTYPE.EQ.1.AND.XTOT.LT..995)WRITE(10,*)' ERROR *****',
C SUM OF X'S = %,XTOT=100.
C IF(NTYPE.EQ.1.AND.XTOT.GT.1.005)WRITE(10,*)' ERROR *****',
C SUM OF X'S = %,XTOT=100.
RETURN

END

***** END OF SUBROUTINE PORT *****
***** SUBROUTINE TRACKER (NUM, IREAD) *****
***** INCLUDE 'COMSUC.F.' *****

C ***** RESET THE RETURNS TO ORIGINAL STATE *****
DO 1300 J=1,NUM

```

84

```

DO 951 I=1,3
DO 951 J=1,1
IF (VAR(1).LE. 0.0 OR. VAR(3).LE. 0.0) GO TO 951
COREL(1,J)=COV(1,J)/(VAR(1)**.5*VAR(J)**.5)
951 CONTINUE
C ***** CALCULATE CUMULATIVE RETURNS *****
PRET(NSTAT)=100
ANIM(NSTAT)=100
DO 569 I=NSTAT+1, IREAD
XX=PRET(I)
YY=ANIM(I)
PRET(I)=PRET(I-1)*(1+XX)
ANIM(I)=ANIM(I-1)*(1+YY)
569 CONTINUE
IF (VAR(1).NE. 0.0) BETA = COV(1,2)/VAR(1)
ALPHA = AVE(2) - BETA*AVE(1)
SST = 0.0
SSE = 0.0
DO 952 I = NSTAT+1, IREAD
SSE = SSE + (ALPHA+BETA*STAT(I,1)-STAT(I,2))**2
SST = SST + (STAT(I,2)-AVE(2))**2
952 CONTINUE
IF (NSINS.GT.2) STDERR = (SSE/DBLE(NSINS-2))**.5
IF (SST.NE.0.0) RSQ = 1 - SSE/SST
C ***** BEGIN WRITING THE TRACKER OUTPUT *****
318 WRITE(10,*) ' TRACKER OUTPUT FILE '
WRITE(10,*) ' MONTH LIA PORT DIF'
WRITE(10,*)
C ***** WRITE MONTH, LIABILITY RET, PORTFOLIO RET, DIFFERENCE *****
DO 700 I=NSTAT+1, IREAD
ISTAT = I-(NSTAT+1)
WRITE(10,612) ISTAT+NM, (STAT(I,J),J=1,3)
612 FORMAT(1X,3X,14,3F13.10)
700 CONTINUE
C ***** WRITE CUMULATIVE RETURNS *****
WRITE(10,*)
WRITE(10,886)
886 FORMAT(1X,'CUMULATIVE VALUES OF LIABILITIES AND ASSETS'/.
/,' MONTH LIABILITIES PORTFOLIO',/)

```

```

DO 147 I=NSTAT, IREAD
ISTAT = I-NSTAT-1
WRITE(10,328) ISTAT+NM, ANIM(1), PRET(1)
328 FORMAT(1X,15,7X,F7.3,10X,F7.3)
847 CONTINUE
837 CONTINUE
C ***** WRITE STATISTICS BASED ON SIMULATIONS MONTHS *****
WRITE(10,*)
WRITE(10,985) NM, NM*NSINS-1
985 FORMAT(/,' STATISTICS SUMMARY '
/,' STATISTICS BASED ON MONTHS ',14,' THROUGH',14)
C ***** WRITE ALPHA, BETA, STDERR, CORREL, AND R-SQUARED *****
WRITE(10,*)
WRITE(10,987) 'ALPHA', ALPHA, 'ANNUAL', ALPHA*1200
WRITE(10,988) 'BETA', BETA
WRITE(10,988) 'STD-ERR', STDERR, 'STDERR*100'
WRITE(10,988) 'CORRELATION', COREL(2,1)
WRITE(10,988) 'R-SQUARED', RSQ, 'RSQ*100'
987 FORMAT(1X,A12,10X,F10.6,10X,F10.2,' %')
988 FORMAT(1X,A12,10X,F10.6,8X,F10.2,' %')
989 FORMAT(1X,A12,10X,F10.6)
C ***** WRITE AVERAGES AND VARIANCES OF 1-LIAB, 2-PORT, 3-DIFFS **
WRITE(10,*)
WRITE(10,*)
WRITE(10,*) 'ANNUAL RETURNS: MEAN STD DEV'
WRITE(10,*)
DO 990 I = 1,3
A/E(I) = (AVE(I)*12)**100
VAR(I) = ((VAR(I)*12)**.5)*100
990 CONTINUE
WRITE(10,916) 'LIABILITY', AVE(1), VAR(1)
WRITE(10,916) 'PORTFOLIO', AVE(2), VAR(2)
WRITE(10,916) 'DIFFERENCE', AVE(3), VAR(3)
916 FORMAT(1X,3X,A11,9X,F9.2,' %',10X,F9.2,' %')
C ***** WRITE PARAMETERS FOR THE RUN *****
WRITE(10,*)
WRITE(10,*)
WRITE(10,*) 'PARAMETERS FOR THIS RUN WERE: '
WRITE(10,*)
WRITE(10,101) TITLE
WRITE(10,102) FILEOUT

```

✱ ✱

2

87

```

SIC(8) - 'FORESTRY'
SIC(9) - 'FISHING HUNTING AND TRAPPING'
SIC(10) - 'METAL MINING'
SIC(11) - 'ANTHRACITE MINING'
SIC(12) - 'BITUMINOUS COAL & LIGNITE MIN.'
SIC(13) - 'OIL & GAS EXTRACTION'
SIC(14) - 'MIN. & QUARRY. OF NONMET. MIN.'
SIC(15) - 'BUILT. CONST.-G. C. & O. B.'
SIC(16) - 'CONST. OTH THAN B.C.-G.C.'
SIC(17) - 'CONST.-SPEC. TRADE CONT.'
SIC(18) - 'FOOD KINDRED PRODUCTS'
SIC(19) - 'TOBACCO MANUFACTURERS'
SIC(20) - 'TEXTILE MILL PRODUCTS'
SIC(21) - 'APP. & OTH. FIN. PROD. MFAOSH'
SIC(22) - 'LUMBER & WOOD PROD. EX. FURN.'
SIC(23) - 'FURNITURE AND FIXTURES'
SIC(24) - 'PAPER AND ALLIED PRODUCTS'
SIC(25) - 'PRINTING PUBLISHING AND A.P.'
SIC(26) - 'CHEMICAL AND ALLIED PROD.'
SIC(27) - 'PETROLEUM REFIN. & REL. PROD.'
SIC(28) - 'RUBBER AND MISC. PLAST. PROD.'
SIC(29) - 'LEATHER & LEATHER PRODUCTS'
SIC(30) - 'STONE CLAY GLASS & CONC. PROD.'
SIC(31) - 'PRIMARY METAL INDUSTRIES'
SIC(32) - 'FAB. METAL PROD. EX. M.A.T.E.'
SIC(33) - 'MACHINERY EXCEPT ELECTRICAL'
SIC(34) - 'ELE. AND ELE. MACH.'
SIC(35) - 'TRANSPORTATIONS EQUIPMENT'
SIC(36) - 'MEAS. ANAL. & CONT. INST. ETC.'
SIC(37) - 'MISC. MANUFACTURING IND.'
SIC(38) - 'RAILROAD TRANS.'
SIC(39) - 'LOC. AND SUB. TRANS. & HPT.'
SIC(40) - 'MOTOR FREIGHT TRANS. AND WARE.'
SIC(41) - 'U.S. POSTAL SERVICE'
SIC(42) - 'WATER TRANSPORTATION'
SIC(43) - 'TRANSPORTATION BY AIR'
SIC(44) - 'PIPE LINES EX. NATURAL GAS'
SIC(45) - 'TRANSPORTATION SERVICES'
SIC(46) - 'COMMUNICATION'
SIC(47) - 'ELECTRIC GAS AND SANIT. SERV.'
SIC(48) - 'WHOLESALE TRADE-DURABLE GOODS'
SIC(49) - 'WHOLESALE TRADE-NONDUR. GOODS'
SIC(50) - 'BUILDING MAT. HARD. GAR. SUPP.'
SIC(51) - 'GENERAL MERCH. STORES'
SIC(52) - 'FOOD STORES'
SIC(53) - 'AUTOMOTIVE DEAL AND GAS. SS'
SIC(54) - 'APPAREL AND ACCESS. STORES'
SIC(55) - 'HOME FURN. AND EQUIP. STORES'
SIC(56) - 'EATING AND DRINKING PLACES'
SIC(57) - 'MISCELLANEOUS RETAIL'
SIC(58) - 'BANKING'
SIC(59) - 'CRED. AGEN. OTH. THAN BANKS.'
SIC(60) - 'FORESTRY'
SIC(61) - 'FISHING HUNTING AND TRAPPING'
SIC(62) - 'SEC. AND COMM. BROK. DEEXSE'
SIC(63) - 'INSURANCE CARRIERS'
SIC(64) - 'INS. AGENTS BROK. SERV.'
SIC(65) - 'REAL ESTATE'
SIC(66) - 'COMB. RE INS. LOANS & LAW OFF.'
SIC(67) - 'HOLD. AND OTHER INV. COMP.'
SIC(68) - 'HOTELS ROOM. HOUSES CAMP AOLP'
SIC(69) - 'PERSONAL SERVICES'
SIC(70) - 'BUSINESS SERVICES'
SIC(71) - 'AUTO. REPAIR SERV. AND GAR.'
SIC(72) - 'MISC. REPAIR SERVICES'
SIC(73) - 'MOTION PICTURES'
SIC(74) - 'AMUSE. AND REC. SERV. EX. MP'
SIC(75) - 'HEALTH SERVICES'
SIC(76) - 'LEGAL SERVICES'
SIC(77) - 'EDUCATIONAL SERVICES'
SIC(78) - 'SOCIAL SERVICES'
SIC(79) - 'MUSEUM ART GALL. BOT. Z.G.'
SIC(80) - 'NONPROFIT MEM. ORGAN.'
SIC(81) - 'MISCELLANEOUS SERVICES'
SIC(82) - 'EXEC. LEG. & GOVT EX. FIN.'
SIC(83) - 'JUSTICE PUBLIC SAFETY'
SIC(84) - 'PUR. FIN. TAX. & MON. POLICY'
SIC(85) - 'ADMIN. OF HUMAN RES. PROGRAMS'
SIC(86) - 'ADMIN. OF QUAL. & HOUS. PROG.'
SIC(87) - 'ADMIN. OF ECONOMIC PROGRAMS'
SIC(88) - 'NATIONAL SEC. & INTL AFFAIRS'
SIC(89) - 'NONCLASSIFIABLE ESTABLISHMENTS'
SIC(90) - 'RETURN'
SIC(91) - 'END'
SIC(92) - 'C. UPDATED December 6, 1988'
SIC(93) - 'END OF SUBROUTINE INDUST'
SIC(94) - 'SUBROUTINE INTRO'
SIC(95) - 'THIS ROUTINE INTRODUCES THE SPANNING PROGRAM AND'
SIC(96) - 'ASKS FOR SCREEN OR FILE INPUT OPTION'
SIC(97) - 'INCLUDE 'CONSUC.F.'
SIC(98) - 'INTEGER RESPON'
SIC(99) - 'PARAMETER (MAXTXT = 50, LINES = 24)'
SIC(100) - 'CHARACTER HEAD*(MAXTXT), LAST*(MAXTXT), OPTION(LINES)*(MAXTXT)'
SIC(101) - 'HEAD - ' WELCOME TO THE NISA SPANNING OPTIMIZER'
SIC(102) - 'NBROPT = 2'
SIC(103) - 'OPTION (1) - 'SINGLE RUN WITH WEIGHTS IN FILE: OLD'
SIC(104) - 'OPTION (2) - 'UPDATE RUN WITH WEIGHTS CALCULATED FROM HOLDINGS'

```

100  
101  
102  
103  
104

4

```
*****  
END *****  
***** END OF SUBROUTINE REDSCRN *****  
*****  
***** SUBROUTINE MODIFY *****  
*****  
*****
```

```

** THIS ROUTINE MODIFIES THE INPUT DATA IF NECESSARY
**
INCLUDE 'CONSUC.F'

WRITE(*,100)(' ',1 = 1,5)
WRITE(*,*)***** CURRENTLY SELECTED PARAMETERS *****
***** DISPLAY SELECTED PARAMETERS *****
CALL DISPLAY
IRESPON = 0
WRITE(*,120)
READ(*,112), END = 20, ERR = 10) IRESPON
IF(IRESPON.EQ.0) THEN
  RETURN
ELSE IF (IRESPON.EQ.1) THEN
  WRITE(*,121)
  READ(*,130), END = 201, ERR = 201) TITLE
ELSE IF (IRESPON.EQ.2) THEN
  WRITE(*,122)
  READ(*,130), END = 202, ERR = 202) FILEOUT
ELSE IF (IRESPON.EQ.3) THEN
  WRITE(*,123)
  READ(*,130), END = 203, ERR = 203) LIABIN
ELSE IF (IRESPON.EQ.4) THEN
  WRITE(*,124)
  READ(*,130), END = 204, ERR = 204) STOCKIN
ELSE IF (IRESPON.EQ.5) THEN
  WRITE(*,125)
  READ(*,130), END = 205, ERR = 205) BONDIN
ELSE IF (IRESPON.EQ.6) THEN
  WRITE(*,126)
  READ(*,130), END = 206, ERR = 206) MMSAVE
ELSE IF (IRESPON.EQ.7) THEN
  WRITE(*,127)
  READ(*,130), END = 207, ERR = 207) NSIMS
ELSE IF (IRESPON.EQ.8) THEN
  WRITE(*,128)
  READ(*,130), END = 208, ERR = 208) NSTOCKS
ELSE IF (IRESPON.EQ.9) THEN
  WRITE(*,129)
  READ(*,130), END = 209, ERR = 209) MNBONDS
ELSE IF (IRESPON.EQ.10) THEN
  WRITE(*,130)
  READ(*,130), END = 210, ERR = 210) IDUMMY
ELSE IF (IRESPON.EQ.11) THEN
  WRITE(*,131)
  READ(*,130), END = 211, ERR = 211) STOCKMIN
ELSE IF (IRESPON.EQ.12) THEN
  WRITE(*,132)
  READ(*,130), END = 212, ERR = 212) YIMAX
ELSE IF (IRESPON.EQ.13) THEN
  WRITE(*,133)
  READ(*,130), END = 213, ERR = 213) SMAX
ELSE IF (IRESPON.EQ.14) THEN
  WRITE(*,134)
  READ(*,130), END = 214, ERR = 214) NSTAT
ELSE IF (IRESPON.EQ.15) THEN
  WRITE(*,135)
  READ(*,130), END = 215, ERR = 215) MRETS
ELSE IF (IRESPON.EQ.16) THEN
  WRITE(*,136)
  READ(*,130), END = 216, ERR = 216) BULLET
ELSE IF (IRESPON.EQ.17) THEN
  WRITE(*,137)
  READ(*,130), END = 217, ERR = 217) TARGET
ELSE IF (IRESPON.EQ.18) THEN
  WRITE(*,138)
  READ(*,130), END = 218, ERR = 218) TURN
END IF
GO TO 10
RETURN
FORMAT('0',A1)
FORMAT('0',) ENTER NUMBER TO MODIFY <ENTER> TO CONTINUE: (,5)
FORMAT('0',) ENTER SPANNING RUN TITLE (,5)
FORMAT('0',) ENTER FILENAME FOR OUTPUT (,5)
FORMAT('0',) ENTER LIABILITY STREAM FILE (,5)
FORMAT('0',) ENTER STOCK RETURN FILE (,5)
FORMAT('0',) ENTER BOND RETURN FILE (,5)
FORMAT('0',) ENTER MONTH TO BEGIN SIMULATION (,5)
FORMAT('0',) ENTER NUMBER OF MONTHS TO SIMULATE (,5)

```



\*\*\*\*\*

2

93

```

C WS      --- WORK SPACE ARRAY OF SIZE AT LEAST
C C      2*(MO+NO)**2 + 12*(MO+NO) + 16 IF ANY EQUALITY CONSTRAINTS
C C      ARE PRESENT OR 2*(MO+NO)**2 + 8*(MO+NO) + 6
C C      IF ALL CONSTRAINTS ARE INEQUALITIES.
C C
C C      IMPLICIT REAL (A-H,O-Z)
C
C DIMENSION A(1),KT(1),RHS(1),COST(1),QUAD(1),INPUT(1),
C 1 TOL(1),X(1),RC(1),DUAL(1),SLK(1),IOUT(1),WS(1)
C CHARACTER*64 TITLE,PFILE
C
C INTEGER ROW,COL,INDX,MORE,P,P1,PSQ
C COMMON /QPCB1/ INPUTS(14),IOUTS(2),ROW,COL,INDX,MORE,P,P1,PSQ,
C 1 NCALL,LENREQ,IO1,IO2,IO3
C EQUIVALENCE (INPUTS(1),NL), (INPUTS(2),NL), (INPUTS(3),NO),
C 1 (INPUTS(4),NO), (INPUTS(5),MINMAX), (INPUTS(6),LENWS),
C 2 (INPUTS(7),MAXIT), (INPUTS(8),KOBJ), (INPUTS(9),JIT),
C 3 (INPUTS(10),JDATA), (INPUTS(11),JPivot), (INPUTS(12),JSOL),
C 4 (INPUTS(13),JOUT), (INPUTS(14),JWIDTH),
C 5 (IOUTS(1),IERR), (IOUTS(2),ITCNT)
C
C COMMON /QPCBR/ TOLS(2)
C EQUIVALENCE (TOLS(1),TZERO), (TOLS(2),TPIV)
C
C DOUBLE PRECISION PIVOT
C COMMON /QPCBD/ PIVOT
C
C CHARACTER*64 TITLES, PFILES
C COMMON /QPCBC/ TITLES, PFILES
C
C DATA NCALL / 0 /
C NCALL = 0
C
C OUTPUT UNITS FOR LINE PRINTER (TERMINAL) AND OUTPUT FILE
C DATA LUPRNT, LUFIL / 6, 92 /
C
C INITIALIZATIONS
C IO1 = LUPRNT
C IERR = 0
C ICIT = 0
C TZERO = TOL(1)
C TP1 = TOL(2)
C NCALL = NCALL + 1
C DO 10 I = 1, 14
C   INPUTS(I) = INPUT(1)
C 10 CONTINUE
C TITLES =

```

```

C INPUT --- FIXED CONSTANTS VECTOR OF LENGTH 21 CONTAINING ...
C C      1 ROW DIMENSION OF A IN CALLING PROGRAM. MINIMUM IS NO.
C C      2 ROW DIMENSION OF QUAD IN CALLING PROGRAM. MINIMUM IS NO.
C C      3 NUMBER OF CONSTRAINTS.
C C      4 NUMBER OF VARIABLES.
C C      5 -0 IF OBJECTIVE IS TO BE MINIMIZED,
C C      -1 IF OBJECTIVE IS TO BE MAXIMIZED.
C C      6 DIMENSION OF WS IN CALLING PROGRAM. MINIMUM IS
C C      2*(MO+NO)**2 + 12*(MO+NO) + 16 IF ANY EQUALITY CONSTRAINTS
C C      ARE PRESENT OR 2*(MO+NO)**2 + 8*(MO+NO) + 6
C C      IF ALL CONSTRAINTS ARE INEQUALITIES.
C C      7 PIVOT LIMIT.
C C      8 COMPUTE OBJECTIVE AT SOLUTION OPTION. -0 NO, -1 YES.
C C      9 OUTPUT PROBLEM PARAMETERS OPTION. -0 NO, -1 YES.
C C      10 OUTPUT PROBLEM DATA OPTION. -0 NO, -1 YES.
C C      -1 OUTPUT WITH A AND QUAD MATRICES IN DENSE FORM,
C C      -2 OUTPUT WITH A AND QUAD MATRIX IN SPARSE FORM.
C C      11 OUTPUT PIVOT INFORMATION. -0 NO, -1 YES.
C C      12 OUTPUT SOLUTION REPORT OPTION. -0 NO, -1 YES.
C C      13 -1 OUTPUT BRIEF REPORT, -2 OUTPUT FULL REPORT.
C C      -0 PRINT ALL OUTPUT, -1 PRINT PROBLEM PARAMETERS AND FINAL
C C      OUTPUT AND FILE OTHER OUTPUT, -2 FILE ALL OUTPUT
C C      14 MAXIMUM WIDTH OF OUTPUT LINES. A MINIMUM OF 72 AND A
C C      MAXIMUM OF 132 WILL BE USED.
C
C TOL --- TOLERANCE VECTOR OF SIZE 2 CONTAINING...
C C      1 ROUND-OFF OR ZERO TOLERANCE. IF .LE. 0 RESET TO 1.E-7
C C      2 PIVOT TOLERANCE. IF .LE. 0 RESET TO 1.E-6
C
C TITLE --- TITLE PRINTED IN OUTPUT. TYPE CHARACTER.
C C      MAY CONTAIN UP TO 64 CHARACTERS.
C C      TRUNCATED TO 64 CHARACTERS IF LONGER.
C C      PFILE --- FILE NAME QUADPR WRITES OUTPUT ONTO IF REQUESTED.
C C      TYPE CHARACTER. MAY CONTAIN UP TO 64 CHARACTERS.
C C      IF FILE CANNOT BE OPENED, A FILE WILL BE CREATED.
C C      AND A MESSAGE PRINTED. IF THE NAME IS ALL BLANKS,
C C      OUTPUT TO BE FILED WILL BE DISCARDED.
C C      OBJECTIVE VALUE.
C C      OBJ --- NO-VECTOR THAT WILL CONTAIN THE SOLUTION.
C C      X --- NO-VECTOR THAT WILL CONTAIN THE REDUCED COSTS.
C C      RC --- NO-VECTOR THAT WILL CONTAIN THE DUAL SOLUTION.
C C      DUAL --- NO-VECTOR THAT WILL CONTAIN THE DUAL SOLUTION.
C C      SLK --- NO-VECTOR THAT WILL CONTAIN THE SLACK VALUES.
C C      IOUT --- VECTOR OF SIZE 2, THAT WILL CONTAIN...
C C      1 STATUS CODE FROM 1 TO 6.
C C      2 - NO SOLUTION.
C C      3 - PIVOT LIMIT REACHED.
C C      4 - INVALID QUAD MATRIX.
C C      5 - PROBLEM DATA ERROR.
C C      6 - ALGORITHM ERROR.
C C      ITCNT 2 NUMBER OF PIVOTS.

```

```

OUTPUT PROBLEM PARAMETERS AND PROBLEM DATA IF REQUESTED.
  IF (JIT.NE.O) CALL QRPRT1
  IF (JDA(A.NE.O) CALL QRPRT2 (A,ML,KT,RHS,COST,QUAD,NL)
OUTPUT HEADER FOR INTERMEDIATE OUTPUT, IF ANY EXPECTED
  IF (JPI/OT.NE.O) CALL QRPRT3
CONSTRUCT INITIAL TABLEAU FOR QUADRATIC PROGRAMMING.
CALL QDPIEP (A,ML,KT,RHS,COST,QUAD,NL,NS,P)
PERFORM PRINCIPAL PIVOTING TO FINAL TABLEAU.

```

95

```

      IF (TZERO .LE. 0.0) TZERO = 1.0E-7
      IF (TPIV .LE. 0.0) TPIV = 1.0E-6
      IF (JWIDTH .LT. 72) JWIDTH = 72
      IF (JWIDTH .GT. 132) JWIDTH = 132
      ITITLE = 0

      CHECK IT
      DO 10 I = 1, MO
      IF (ABS(KT(I)) .GT. 1) GO TO 20
10 CONTINUE
      GO TO 10
20 CONTINUE
      WRITE (101,22) NCALL
22 FORMAT (///5X,QUADPR CALL',14)
      WRITE (101,24) (KT(I),I=1,MO)
24 FORMAT (//1X,*** ERROR, ILLEGAL VALUES IN KT. KT CONTAINS'
1 / (6I12) )
      IERR = 5
      ITITLE = 1

      CHECK INPUT
30 CONTINUE
      ERROR(1) = (MO .LT. 0 .OR. MO .GT. ML)
      ERROR(2) = (MO .LT. 1 .OR. MO .GT. NL)
      ERROR(3) = (LENMS .LT. LENREQ)
      DO 33 I = 1, 4
      J = ZERONE(I)
      ERROR(I+3) = (INPUTS(J) .NE. 0 .AND. INPUTS(J) .NE. 1)
33 CONTINUE
      DO 34 I = 1, 3
      J = ZERO12(I)
      ERROR(I+7) = (INPUTS(J) .LT. 0 .OR. INPUTS(J) .GT. 2)
34 CONTINUE
      DO 35 I = 1, 10
      IF (ERROR(I)) GO TO 38
35 CONTINUE
      GO TO 40
38 CONTINUE
      IERR = 5
      IF (ITITLE .EQ. 0) WRITE (101,22) NCALL
      WRITE (101,40) (INPUTS(I),I=1,14)
40 FORMAT (//1X,*** ERROR, INCONSISTENT DATA IN INPUT.'
1 / (10I12) / 5112 / 5112 / 4112 /
2 / 5X 'THE FOLLOWING ERRORS OCCURRED.' / )
      IF (ERROR(1)) WRITE (101,41)
41 FORMAT (5X, INPUT(3) NEGATIVE OR EXCEEDS INPUT(1)')
      IF (ERROR(2)) WRITE (101,42)
42 FORMAT (5X, INPUT(4) NON-POSITIVE OR EXCEEDS INPUT(2)')
      IF (ERROR(3)) WRITE (101,43) LENREQ
43 FORMAT (5X, INPUT(6) TOO SMALL. FOR THIS PROBLEM,')

```

```

      CALL QDCOMP (MS,WS(L1),WS(L2),WS(L3),WS(L4))
      GET SOLUTION AND OBJECTIVE VALUE IF REQUESTED.
      CALL QDSOLN (KT,COST,QUAD,NL,MS,P,WS(L4),X,RC,DUAL,SLK,
1 MS(L1),OBJ)
      OUTPUT SOLUTION REPORT IF REQUESTED.
      IF (JSOL .NE. 0) CALL QRPRI6 (X,RC,DUAL,SLK,OBJ)
      SET EXIT VALUES, IOUT.
200 CONTINUE
      IOUT(1) = IERR
      IOUT(2) = ITCNT
      CLOSE PRINT FILE IF REQUIRED.
      IF (IOUT .ST. 0) CLOSE (LUFIL,STATUS='KEEP')
      RETURN
      END
*****
      SUBROUTINE QDRVER (KT)
      CHECK INPUT DATA FOR QUADPR
      IMPLICIT REAL (A-H,O-Z)
      DIMENSION KT(1)
      INTEGER ROW,COL,INDX,P,P1,PSQ
      COMMON /QPCB/ INPUTS(14), IOUTS(2), ROW,COL,INDX,MORE,P,P1,PSQ,
1 NCALL,LENREQ,IO1,IO2,IO3
      EQUIVALENCE (INPUTS(1),ML), (INPUTS(2),NL), (INPUTS(3),MO),
2 (INPUTS(4),NO), (INPUTS(5),MINMAX), (INPUTS(6),LENMS),
3 (INPUTS(7),MAXIT), (INPUTS(8),KOBJ), (INPUTS(9),JIT),
4 (INPUTS(10),JDATA), (INPUTS(11),JPIVOT), (INPUTS(12),JSOL),
5 (IOUTS(1),IERR), (IOUTS(2),ITCNT)
      COMMON /QPCBR/ TOLS(2)
      EQUIVALENCE (TOLS(1),ZERO), (TOLS(2),TPIV)
      INTEGER ZERONE(4),ZERO12(3)
      DATA ZERONE / 5,8,9,11 /
      DATA ZERO12 / 10,12,13 /
      ITITLE = 0
      DEFAULT VALUES
      IF (MAXIT .LE. 0) MAXIT = 1000

```

96

```

C
MIN. PROBLEM
B(1,1) = 2.0*QUAD(1,1)
B(1,P1) = COST(1)
IF (NO.EQ.1) GO TO 110
DO 10 K = 2, NO
  I(K,K) = 2.0*QUAD(K,K)
  I(K,P1) = COST(K)
10 CONTINUE
DO 10 L = 1, N1
  L1 = L + 1
  IO 20 K = L1, NO
  B(K,L) = QUAD(K,L) + QUAD(L,K)
20 CONTINUE
GO TO 80
MAX. PROBLEM
C
40 CONTINUE
B(1,1) = -2.0*QUAD(1,1)
B(1,P1) = -COST(1)
IF (NO.EQ.1) GO TO 110
DO 10 K = 2, NO
  E(K,K) = -2.0*QUAD(K,K)
  E(K,P1) = -COST(K)
50 CONTINUE
DO 70 L = 1, N1
  L1 = L + 1
  D) 60 K = L1, NO
  B(K,L) = -QUAD(K,L) - QUAD(L,K)
70 CONTINUE
C
80 CONTINUE
FILL UPPER TRIANGLE OF [1].
DO 100 L = 2, NO
  L1 = L - 1
  D) 90 K = 1, L1
  B(K,L) = B(L,K)
90 CONTINUE
100 CONTINUE
C
CHECK FOR CASE OF NO CONSTRAINTS.
IF (IO.EQ.0) GO TO 240
110 CONTINUE
C
FILL [5].
N1 = NO + 1
DO 120 L = N1, P
  D) 120 K = N1, P
  B(K,L) = 0.0
120 CONTINUE
130 CONTINUE
C

1 ' MUST BE AT LEAST', 18)
DO 45 I = 1, 4
  J = ZERO(I)
  IF (ERROR(I+3)) WRITE (101,46) J
45 CONTINUE
46 FORMAT (5X,'INPUT(',12,') NOT 0 OR 1')
DO 47 I = 1, 3
  J = ZERO12(I)
  IF (ERROR(I+7)) WRITE (101,48) J
47 CONTINUE
48 FORMAT (5X,'INPUT(',12,') NOT 0, 1, OR 2')
50 CONTINUE
RETURN
END
*****
SUBROUTINE QDPREP (A,LDA,KT,RHS,COST,QUAD,LQ,B,LDB)
CONSTRUCT THE INITIAL TABLEAU FOR QUADRATIC PROGRAMMING.
IT HAS THE FORM
      QUAD*QUAD (1)
      [1] : A : COST
      B : [2] : [3]
      A : 0 : -RHS
      [4] : [5] : [6]
      *****
WHERE A', RHS', COST', QUAD' ARE OBTAINED FROM A, RHS, COST, QUAD BY
CONVERTING THE PROBLEM TO MINIMIZATION WITH ALL .GE. CONSTRAINTS.
IMPLICIT REAL (A-H,O-Z)
DOUBLE PRECISION B
DIMENSION A(LDA,1),KT(1),RHS(1),COST(1),QUAD(LQ,1),B(LDB,1)
INTEGER ROW,COL,INDX,MORE,P,P1,PSQ,
COMMON /QPRCB1/ INPUTS(14),IOUTS(2),ROW,COL,INDX,MORE,P,P1,PSQ,
1 NCALL,LENREQ,IO1,IO2,IO3
EQUIVALENCE (INPUTS(1),ML), (INPUTS(2),ML), (INPUTS(3),MO),
1 (INPUTS(4),MO), (INPUTS(5),MINMAX), (INPUTS(6),LEWMS),
2 (INPUTS(7),MAX1), (INPUTS(8),KOB), (INPUTS(9),JIT),
3 (INPUTS(10),JDATA), (INPUTS(11),JPIVOT), (INPUTS(12),JSOL),
4 (INPUTS(13),JOUT), (INPUTS(14),JWIDTH),
5 (IOUTS(1),IERR), (IOUTS(2),ITCNT)
C
FILL LOWER TRIANGLE OF [1] AND [3].
N1 = NO - 1
IF (MINMAX.EQ.1) GO TO 40

```

97

```

C      COMPUTE THE FINAL TABLEAU FOR QUADRATIC PROGRAMMING GIVEN
C      THE INITIAL TABLEAU USING THE METHOD OF PRINCIPAL PIVOTING.
C      IMPLICIT REAL (A-H,O-Z)
C      DOUBLE PRECISION B,TEMP,U
C      INTEGER JZ,JW
C      DIMENSION B(1),TEMP(1),U(1),JZ(1),JW(1)
C
C      INTEGER ROW,COL,INDX,P,P1,PSQ
C      COMMON /QPRCB1/ INPUTS(14),IOUTS(2),ROW,COL,INDX,MORE,P,P1,PSQ,
1      ICALL,LENREQ,IOI,IOZ,IO3
C      EQUIVALENCE (INPUTS(1),ML), (INPUTS(2),ML), (INPUTS(3),MO),
1      INPUTS(4),NO), (INPUTS(5),MINMAX), (INPUTS(6),LEHMS),
2      INPUTS(7),MAXIT), (INPUTS(8),KOB), (INPUTS(9),JIT),
3      INPUTS(10),JDATA), (INPUTS(11),JPivot), (INPUTS(12),JSOL),
4      INPUTS(13),JOUT), (INPUTS(14),JWIDTH), (IOUTS(1),ITCNT),
5      IOUTS(2),ITCNT)
C
C      COMMON /QPRCB2/ TOLS(2)
C      EQUIVALENCE (TOLS(1),TZERO), (TOLS(2),TPIV)
C
C      DOUBLE PRECISION PIVOT
C      COMMON /QPRCB3/ PIVOT
C
C      DOUBLE PRECISION Z,BS,BST
C      CHARACTER*48 MSG(4)
C
C      DATA MSG / 'ALGORITHM ERROR, INDEX VALUE NON-DECREASING.
2      'NO SOLUTION, INVALID QUADRATIC COSTS MATRIX.
3      'NO SOLUTION, INFEASIBLE OR UNBOUNDED PROBLEM.
4      'ERROR, ITERATION LIMIT REACHED.
C
C      *****INITIALIZATIONS
C      ITCN = 0
C      JFLAG = 0
C      INDX = P1
C      *****BASIC VARIABLES GET NEGATIVE VALUES.
C      *****NONBASIC VARIABLES(JZ) GET POSITIVE VALUES.
C      DO 30 I = 1, P
C      JZ(I) = -1
C      JZ(1) = -1
C      30 CONTINUE
C      40 ITEM = 0
C      *****CHECK FOR NONDECREASING INDEX (ALLOW UP TO 5 EQUAL INDEXES FOR
C      TOLERANCE PURPOSES).
C      IJ = JSQ
C      DO 50 I = 1, P
C      IJ = IJ + 1
C      IF (B(IJ).LT. (-TZERO)) ITEM = ITEM + 1
C      50 CONTINUE
C
C      IF THERE ARE R .EQ. CONSTRAINTS, EXPRESS AS R+1 .GE. CONSTRAINTS,
C      THE LATTER BEING MINUS THE SUM OF THE OTHERS.
C      DO 140 L = 1, NO
C      B(P,L) = 0.0
C      140 CONTINUE
C      B(P,P) = 0.0
C
C      FILL [4] AND [6].
C      150 CONTINUE
C      DO 210 I = 1, MO
C      K = NO + I
C      IF (KT(I).LT. 0) GO TO 170
C      IF (KT(I).GT. 0) GO TO 190
C      .EQ. CONSTRAINT
C      DO 160 L = 1, NO
C      B(K,L) = A(I,L) - B(K,L)
C      B(P,L) = B(P,L) - B(K,L)
C      160 CONTINUE
C      B(K,P) = - RHS(I)
C      B(P,P) = B(P,P) - B(K,P)
C      GO TO 210
C      .GE. CONSTRAINT
C      170 CONTINUE
C      DO 180 L = 1, NO
C      B(K,L) = A(I,L)
C      180 CONTINUE
C      B(K,P) = - RHS(I)
C      GO TO 210
C      .LE. CONSTRAINT
C      190 CONTINUE
C      DO 200 L = 1, NO
C      B(K,L) = - A(I,L)
C      200 CONTINUE
C      B(K,P) = RHS(I)
C      210 CONTINUE
C
C      FILL [2].
C      DO 230 L = 1, NO
C      DO 220 K = M1, P
C      B(L,K) = - B(K,L)
C      220 CONTINUE
C      230 CONTINUE
C
C      FINISHED
C      240 CONTINUE
C      RETURN
C      END
C      *****
C      SUBROUTINE QDCOMP (B,TEMP,U,JZ,JW)

```

98

```

IF (B(KJ)) .LT. (-TZERO) .OR. BST .GE. (-TPIV)) GO TO 140
IF (DABS(B(KJ)) .LT. TZERO .AND. B(IJ) .LT. (-TZERO)) GO TO 120

IF (B(KJ)) 130, 140, 130
ROI = J
GO TO 150
130 BS = -BST / B(KJ)
IF (BST .LE. BS) GO TO 140
BS = BST
ROI = J
140 CONTINUE
IE (ROW .EQ. 0) GO TO 210
C*****NONPRINCIPAL PIVOT ON B(S,R)
C*****AT THIS POINT THE BLOCKING VARIABLE IS B(NS,I).
150 I1 = -JW(ROW)
NFLAG = 1
GO TO 500
160 CONTINUE
IF (ITCNT .GE. MAXIT) GO TO 440
ITCNT = ITCNT + 1
C*****ATTEMPT TO PROGRESSIVELY RELAX THE CONVERGENCE PARAMETER
C*****INDICATE THE NUMBER OF ITERATIONS
WRITE(*,101) OPTIMIZATION ITERATION , ITCNT
101 FORMAT(' ', A50, I5)

JFLAG = 0
C*****PUT TO ERANCE ON B(I,P1) AFTER A NONPRINCIPAL PIVOT.
IJ = PSQ + 1
IF (B(IJ) .LT. (-TZERO)) GO TO 170
GO TO 10

C*****THE NEW DRIVING VARIABLE IS THE COMPLEMENT OF THE OLD BLOCKING
170 DO 180 JJ = 1, P
IF (J2(JJ) .EQ. 111) GO TO 190
180 CONTINUE
C*****NO COMPLEMENT VARIABLE (NOT ALLOWED IF POS. SEMIDEF.)
GO TO 410
190 COL = IJ
C*****THE DISTINGUISHED VARIABLE IS STILL THE I-TH ROW.
IJ = P(COL-1) + 1
IF (B(IJ) .GT. TPIV) GO TO 200
II = COL
GO TO 40
200 KJ = PSQ + 1
Z = -B(KJ) / B(IJ)
GO TO 110
C*****PRINCIPAL PIVOT ON B(I,I).
210 LL = ROW
ROW = I
NFLAG = 2

IF (ITEMP .NE. INDX) ICT = 0
IF (ITEMP .EQ. INDX) ICT = ICT + 1
IF (ITEMP .GT. INDX .OR. ICT .EQ. 5) GO TO 400
C*****IF THE INDEX IS ZERO YOU ARE DONE.
IF (ITEMP .EQ. 0) GO TO 300
INDX = ITEMP
IJ = PSQ
DO 60 I = 1, P
IJ = IJ + 1
IF (B(IJ) .LT. (-TZERO)) GO TO 70
60 CONTINUE
70 IJ = P*(I-1) + 1
C*****CHECK FOR NEGATIVE DIAGONAL ELEMENT (NOT ALLOWED IF POS. SEMIDEF.)
IF (B(IJ) .LT. (-TPIV)) GO TO 410
IF (B(IJ) .GT. TPIV) GO TO 100
C*****IF POTENTIAL PIVOT ELEMENT IS NEAR ZERO, USE CODE BELOW
C*****TO DETERMINE THE BLOCKING VARIABLE.
II = I
80 ROW = 0
TMP = 1000000.
IJ = PSQ
KJ = P*(I-1)
DO 90 J = 1, P
IJ = IJ + 1
KJ = KJ + 1
IF (B(IJ) .LT. (-TZERO)) GO TO 90
C*****IF ALL COL. ELEMENTS ARE NONNEGATIVE, THEN THE DRIVING VARIABLE
C*****IS UNBLOCKED AND THERE IS NO SOLUTION.
IF (B(KJ) .GE. (-TPIV)) GO TO 90
Z = -B(IJ) / B(KJ)
IF (Z .GE. TMP) GO TO 90
TMP = Z
ROW = J
90 CONTINUE
IF (ROW .EQ. 0) GO TO 420
COL = IJ
GO TO 150
C*****COMPUTE Z = -Q(R)/B(R,R). THE INCREASE IN THE DRIVING VARIABLE.
100 IJ = PSQ + 1
KJ = P*(I-1) + 1
Z = -B(IJ) / B(KJ)
COL = I
110 BS = 0
ROW = 0
C*****FIND THE NEW BLOCKING VARIABLE. BASED ON WHICH BASIC VARIABLE IS
C*****MOST AFFECTED BY THE CHANGE IN THE DRIVING VARIABLE.
IJ = (COL-1)*P
KJ = PSQ
DO 140 J = 1, P
IJ = IJ + 1
KJ = KJ + 1
BST = B(IJ)*Z + B(KJ)

```



\*\*\*\*\*PIVOT CODE: A NORMAL JORDAN PIVOT.  
\*\*\*\*\*QUADPP USES THE ELEMENTARY MATRIX.

100

```

        GO TO 500
220 CONTINUE
    IF (JFLAG.EQ. 1) GO TO 290
    JFLAG = 1
    C*****MUST REARRANGE THE ROWS AND COLUMNS FOR COMPLEMENTARITY
    C*****EXCHANGE ROWS
230 LLL = 0
    DO 250 I = 1, P
        K = IABS(JW(I))
        IF (K.EQ. 1) GO TO 250
        IJ = K
        KJ = I
        DO 240 J = 1, P1
            Z = B(IJ)
            B(IJ) = B(KJ)
            B(KJ) = Z
        IJ = IJ + P
        KJ = KJ + P
240 CONTINUE
        II = JW(I)
        JW(I) = JW(K)
        JW(K) = II
        LLL = LLL + 1
250 CONTINUE
    IF (LLL.GE. 2) GO TO 230
    C*****EXCHANGE COLUMNS
260 LLL = 0
    DO 280 J = 1, P
        K = IABS(JZ(J))
        IF (K.EQ. 1) GO TO 280
        IJ = P*(K-1)
        KJ = P*(J-1)
        DO 270 I = 1, P
            Z = IJ + I
            IJ = KJ + I
            Z = B(IJ)
            B(IJ) = B(KJ)
            B(KJ) = Z
270 CONTINUE
        II = JZ(J)
        JZ(J) = JZ(K)
        JZ(K) = II
        LLL = LLL + 1
280 CONTINUE
    IF (LLL.GE. 2) GO TO 260
290 ROW = LL
    ITCNT = ITCNT + 1
    C*****ATTEMPT TO PROGRESSIVELY RELAX THE CONVERGENCE PARAMETER
    C
    TP1V = TP1V + TOLS(2)/50.
    C*****INDICATE THE NUMBER OF ITERATIONS
    C*****WRITE(*,101)'OPTIMIZATION ITERATION ',ITCNT

```

```

        IF (JP1VOT.EQ. 1) CALL QRPRTS
        IF (INDX.GT. 1) GO TO 40
    **NORMAL TERMINATION
    DO IEIR = 1
        GO TO 600
    ***ERROR TERMINATION
    DO K = 1
        IEIR = 6
        GO TO 460
    10 K = 2
        IEIR = 4
        GO TO 460
    120 K = 3
        IEIR = 2
        GO TO 460
    440 K = 4
        IEIR = 3
        GO TO 460
    460 WRITE (103,470) MSG(K),ITCNT
    470 FORMAT( / 1X,A4B,16, / PIVOTS PERFORMED. / )
    GO TO 600

```

```

    *****PIVOT CODE:
    *****JADPP USES A NORMAL JORDAN PIVOT.
    *****WHENEVER WE CAN SAVE MEMORY BY USING THE ELEMENTARY MATRIX.
    *****STORE PIVOT ELEMENT
    500 CONTINUE
        JC = P*(COL-1)
        CJ = KC + ROW
        PIVOT = B(KJ)
        Z = 1.000 / PIVOT
        *****STORE ELEMENTARY-VECTOR AND U-VECTOR.
        IJ = ROW
        ISAVE = 1
        DO 510 I = 1, P1
            U(I) = B(IJ)
            IJ = IJ + P
510 CONTINUE
        IJ = KC
        DO 520 I = 1, P
            IJ = IJ + 1
            TEMP(I) = -B(IJ) * Z
520 CONTINUE
        TEMP(ROW) = -Z - 1.000
    C*****COMPUTE ALL COLUMNS EXCEPT PIVOT COLUMN.
        IJ = 0
        DO 550 J = 1, P1
            IF (J.NE. COL) GO TO 530
            IJ = IJ + P

```

101

DOUBLE PRECISION DART, BB

GET J, RC, DUAL, SLK

00 CONTINUE

KART = MO + 1

DO 110 J = 1, NO

X(J) = 0.0

R(J) = 0.0

110 CONTINUE

DO 120 I = 1, MO

DUAL(I) = 0.0

SLK(I) = 0.0

120 CONTINUE

DART = 0.000

DO 170 I = 1, P

BB = B(I,P1)

IF (DABS(BB)) .LT. TZERO) BB = 0.000

J = JW(I)

IF (J .LT. 0) GO TO 150

IF (J .GT. NO) GO TO 130

X(J) = BB

GO TO 170

130 CONTINUE

J = J - NO KART GO TO 140

IF (K .EQ. KART) GO TO 140

DUAL(K) = BB

GO TO 170

140 CONTINUE

DART = BB

GO TO 170

150 CONTINUE

J = -J

IF (J .GT. NO) GO TO 160

X(J) = BB

GO TO 170

160 CONTINUE

C = J - NO

IF (K .EQ. KART) GO TO 170

SLK(K) = BB

GO TO 170

170 CONTINUE

C

C

ADJUST DUALS IF ANY EQUALITIES

IF (MORE .EQ. 0) GO TO 200

DO 190 I = 1, MO

IF (KT(I) .NE. 0) GO TO 190

IF (MINMAX .EQ. 1) GO TO 180

DUAL(I) = DUAL(I) - DART

GO TO 190

CONTINUE

180 DUAL(I) = DART - DUAL(I)

GO TO 550  
 DO 540 I = 1, P  
 IJ = IJ + 1  
 B(IJ) = B(IJ) + U(J)\*TEMP(I)

540 CONTINUE

550 CONTINUE

C\*\*\*\*COMPUTE PIVOT COLUMN.

IJ = KC

DO 560 I = 1, P

IJ = IJ + 1

B(IJ) = -TEMP(I)

560 CONTINUE

C\*\*\*\*COMPUTE NEW PIVOT ELEMENT.

B(KJ) = Z

C\*\*\*\*KEEP TRACK OF BASIC AND NONBASIC VARIABLES

K = JW(ROW)

JW(ROW) = JZ(COL)

JZ(COL) = K

C IF (JPIVOT .EQ. 1) CALL QRPRT4

I = ISAVE

GO TO (160, 220), NFLAG

C\*\*\*\*FINISHED

600 RETURN

END

-----

SUBROUTINE QDSOLN (KT,COST,QUAD,LDQ,B,LDB,JW,X,RC,DUAL,SLK,

TEMP,OBJ)

C

C RETURN SOLUTION TO QUADRATIC PROGRAMMING PROBLEM FROM FINAL

C TABLEAU AND COMPUTE OBJECTIVE VALUE IF REQUESTED.

C CONVERTING THE PROBLEM TO MINIMIZATION WITH ALL .GE. CONSTRAINTS.

C IMPLICIT REAL (A-H,O-Z)

C

DOUBLE PRECISION B,TEMP

C DIMENSION KT(1),COST(1),QUAD(LDQ,1),B(LDB,1),JW(1),X(1),RC(1),

DUAL(1),SLK(1),TEMP(1)

C

C INTEGER ROW,COL,INDX,MORE,P,P1,PSQ

C COMMON /QPCRB/ INPUTS(14),IOUTS(2),ROW,COL,INDX,MORE,P,P1,PSQ,

NCALL,LENREQ,IO1,IO2,IO3

C EQUIVALENCE (INPUTS(1),ML), (INPUTS(2),NL), (INPUTS(3),MO),

1 (INPUTS(4),NO), (INPUTS(5),MINMAX), (INPUTS(6),LENMS),

2 (INPUTS(7),MAXIT), (INPUTS(8),KOBJ), (INPUTS(9),JIT),

3 (INPUTS(10),JDATA), (INPUTS(11),JPIVOT), (INPUTS(12),JSOL),

4 (INPUTS(13),JOUT), (INPUTS(14),JWIDTH),

5 (IOUTS(1),IERR), (IOUTS(2),ITCNT)

C COMMON /QPCRB/ TOLS(2)

C EQUIVALENCE (TOLS(1),TZERO), (TOLS(2),TPIV)

C

```

190 CONTINUE
C
C COMPUTE OBJECTIVE VALUE ONLY IF REQUESTED.
200 CONTINUE
IF (KOBJ .EQ. 0) GO TO 250
OBJ = 0.
DO 220 J = 1, NO
TEMP(1) = 0.000
DO 210 J = 1, NO
TEMP(1) = TEMP(1) + X(J)*QUAD(1,J)
210 CONTINUE
220 CONTINUE
DO 230 J = 1, NO
OBJ = OBJ + TEMP(J)*X(J)
230 CONTINUE
DO 240 J = 1, NO
OBJ = OBJ + COST(J)*X(J)
240 CONTINUE
C
C FINISHED
250 CONTINUE
RETURN
END
C*****
C UPDATED December 6, 1988
C*****
SUBROUTINE QSET(NUM,MAXWS,ITURN)
***** INCLUDE 'CONSUC.F' *****
REAL Q(4),S(4),XMAX(ND)
INTEGER K(4)
A
REAL QC(NDID,ND),QRHS(NDID),QTOL(2),QRC(ND)
QUADAL(NDID),QSLK(NDID),QWS(1:ISIZEQ),QBS(ND)
INTEGER QKT(NDID),INPUT(14),QIOUT(2),NUM,MAXWS
CHARACTER*30 PFILE
C ***** IF THE OPTIMAL PORTFOLIO IS WITHIN TURNOVER LIMIT, RETURN *****
IF (ITURN .EQ. 1) THEN
COST = 0.0
DO 5 I = 1, NUM
COST = COST + ABS(X(I)) - OLDWT(I)
5 CONTINUE
TURN = TURN/2.0 * COST
END IF
DO 10 J = 1, ID
DO 10 J = 1, NUM
QC(I+NUM,J) = C(J,I)
10

```

103

```

      QRHS(1) = OLDWT(1)
      END IF
    END IF
  END IF
  ELSE
    QRHS(1) = D(1-NUM)
    END IF
    QKT(1) = -1
    CONTINUE
  30
    INPUT(1) = ND+ID
    INPUT(2) = ND
    INPUT(3) = NUM+ID
    INPUT(4) = NUM
    INPUT(5) = 0
    INPUT(6) = MAXVS
    INPUT(7) = 0
    INPUT(8) = 0
    INPUT(9) = 0
    INPUT(10) = 0
    INPUT(11) = 0
    INPUT(12) = 0
    INPUT(13) = 0
    INPUT(14) = 0
    QTOL(1) = 0.0
    C E-6 QTOL(2) = 0.000001
    C E-4 QTOL(2) = 0.0001
    PFILE = 'TEMP.Q'
    IF (ITURN.EQ.1) QRHS(NUM+6) = -TURN
    DO 40 1 = 1, NUM
      IF (ITURN.EQ.1) THEN
        C***IF THE OLD WEIGHT EQUALS ZERO
        IF (OLDWT(1) .LT. 0.001) THEN
          QC(NUM+6,1) = 1.0
          QRHS(NUM+6) = QRHS(NUM+6) - OLDWT(1)
        C***IF THE OLD WEIGHT EQUALS THE UPPER BOUND
        ELSE IF (OLDWT(1) .GT. 0.95*BDU(1)) THEN
          QC(NUM+6,1) = 1.0
          QRHS(NUM+6) = QRHS(NUM+6) + OLDWT(1)
        C***IF X(1) IS LESS THAN THE OLD WEIGHT
        ELSE IF (X(1) .LE. OLDWT(1)) THEN
          QC(NUM+6,1) = 1.0
          QRHS(NUM+6) = QRHS(NUM+6) + OLDWT(1)
        C***IF X(1) IS GREATER THAN THE OLD WEIGHT
        ELSE IF (X(1) .GT. OLDWT(1)) THEN

```

104

```

80      XMAX(1) = X(1)
      CONTINUE
      ICOUNT = ICOUNT + 1
      GO TO 49

      RETURN
      END

C ***** December 6, 1988
C ***** SUBROUTINE SUCCESS *****
C ***** INCLUDE COMSUC.F *****

C ***** CALCULATE VARIANCE AND COVARIANCE *****
      XVAR = 0.0
      COVAR = 0.0
      BB = 0.0
      DO 10 I = 1, NUM
        F(I) = 0.0
        COVAR = COVAR + B(I)*X(1)/BB
        DO 20 J = 1, NUM
          XVAR = XVAR + X(1)*X(J)*A(I,J)
          F(I) = F(I) + 2.0*X(J)*A(I,J)
        CONTINUE
      CONTINUE
      DO 30 I = 1, NUM
        G(I) = B(I)/BB*VAR1 - 0.5*COVAR*VAR3*F(I)
        DO 40 J = 1, NUM
          H(I,J) = -0.5*COVAR*(-1.5*VAR3*F(I)*F(J) + VAR3*2.0*A(I,J))
          G(I) = G(I) - X(J)*H(I,J)
          H(I,J) = H(I,J) + X(J)*H(I,J)
        CONTINUE
      CONTINUE
      H(I,J) = 0.5*H(I,J)
      CONTINUE
      SIT F( ) EQUAL TO X( ) TO SAVE THE OLD X( ) *****
      CONTINUE

      DO 50 I = 1, NUM
        F(I) = X(I)
      CONTINUE
      RETURN
      END

C ***** END OF SUBROUTINE SUCCESS *****

47      CONTINUE
      DO 48 I = 1, MAXHS
        QMS(I) = 0.0
      CONTINUE
      C ***** DONE INITIALIZING *****

      CALL QUAD1 (QC,QKT,QRHS,OB,A,INPUT,QTOL,TITLE,
      & PFILE,OBJ,X,QRC,QDUAL,QSLK,QIOUT,QMS)

      C ***** CALCULATE CORRELATION *****

      CONTINUE
      INPUT(5) = 1
      IF(QIOUT(1).NE.1) RETURN
      COVAR = 0.0
      DO 50 I = 1, NUM
        COVAR = COVAR + X(1)*B(I)/BULLET
      CONTINUE

      XVAR = 0.0
      COST = 0.0
      DO 60 I = 1, NUM
        COST = COST + ABS(X(1)-OLDWT(I))
        DO 70 J = 1, NUM
          XVAR = XVAR + X(1)*X(J)*A(I,J)
        CONTINUE
      CONTINUE

      OBJLAST = OBJSTAR
      OBJSTAR = COVAR/(XVAR**5)
      C ***** DONE WITH CORRELATION CALCULATION *****
      CHANGI = (OBJSTAR-OBJMAX)/ABS(OBJMAX)
      WRITE(*,990)ICOUNT,SCALE,OBJSTAR*1000,DELTA*100
      990  FORMAT(' ',13,' SCALE=',F7.2,' OBJ=',F7.2,' ',F7.2,' %')
      991  FORMAT(' ',13,' ASO)
      C ***** TERMINATE IF MOVE IS SMALL *****
      IF (DELTA.LT. CHECK/5.) RETURN
      IF (ICOUNT.GT. 6) RETURN
      WRITE(*,*) 'CALLING HESSIAN'
      CALL SUCCESS (NUM)
      CALL QUAD1 (QC,QKT,QRHS,G,H,INPUT,QTOL,TITLE,
      & PFILE,OBJ,X,QRC,QDUAL,QSLK,QIOUT,QMS)
      IF(QIOUT(1).NE.1) THEN
        DO 75 I = 1, NUM
          X(I) = XMAX(I)
        CONTINUE
      RETURN
      END IF
      C ***** CALCULATE THE CHANGE IN X *****
      DELTA = 0.0
      DO 80 I = 1, NUM
        DELTA = DELTA + ABS(X(I)-F(I))

```

```

*****-----
C *** PARAMETERS ARE USED TO DIMENSION THE COMMON ARRAYS      *
C *** IN EACH SUBROUTINE.                                       *
C ***                                                            *

C *** ND = TOTAL NUMBER OF STOCK WEIGHTS BEING COMPUTED.      *
C *** ID = THE MAXIMUM NUMBER OF CONSTRAINT EQUATIONS.        *
C *** N2D = THE DIMENSION ON WORKING AREAS ( 2 * ND )         *
C *** KEQ = NUMBER OF CONSTRAINTS WHICH WILL BE EQUALITIES    *

C *** NRT = NUMBER OF RETURNS                                  *
C *** NST = USED FOR STATISTICS                                 *

C *** ISIZEQ = 2 * ( ND*2 + ID )**2 + 8*(ND*2+ID) +6          *
C *** NDID = ND + ID                                           *

C *****
C   LAST UPDATED: November 19, 1988
C   IMPLICIT REAL (A-H,O-Z)
C   IMPLICIT INTEGER (I-N)
C   PARAMETER (KEQ = 0)

C 280
C   PARAMETER (ND = 280, ID = 6 , N2D = 560)
C   PARAMETER (NDID = 286, ISIZEQ = 645246)

C 250
C   PARAMETER (ND = 250, ID = 6 , N2D = 500)
C   PARAMETER (NDID = 256, ISIZEQ = 516126)

C 100
C   PARAMETER (ND = 100, ID = 6 , N2D = 200)
C   PARAMETER (NDID = 106, ISIZEQ = 86526)

C   PARAMETER (NRT = 48, NST = 3)
C   PARAMETER (LENGTH = 40)
C   CHARACTER*(LENGTH) FILEOUT,TITLE,NAME(ND),PORTRETS,GARB,ANSWER,
&   OLD,LIANAME,LIABIN,STOCKIN,BONDIN,IDC(ND),SIC(99)

COMMON /ARRY/ A(ND,ND),C(ND,ID), H(ND,ND), G(ND), F(ND)
COMMON /VECT/ B(ND),D(ID),BDL(ND),BDU(ND),X(ND),OLDWT(ND)
COMMON /RETS/ RET(NRT,ND),ANIM(NRT),PRET(NRT),ISIC(ND),AVE(ND)
COMMON /TRAC/ COV(NST,NST),STAT(NRT,NST),COREL(NRT,NST)
COMMON /TRA2/ PRICE(ND),VOL(ND),CUM(ND),VAR(3),LIANAME,TITLE
COMMON /RTNIN/FILEOUT,LIABIN,STOCKIN,BONDIN,IDC,NAME,SIC
COMMON /DAT/  NMSAVE,NM,NSIMS,NSTOCKS,NBONDS,NSTAT,NRETS,IDUMMY,
&   IHOLD
&   COMMON /MAX/ STOCKMIN,YIMAX,SMAX,SCALE,BRET,XFACTOR,XBUPPER,
&   XBLOWER,BULLET,TARGET,TURN,BULL1,TARG1,TURN1,PORTVAL
C END OF COMMON.F

```

106

APPENDIX IV

"FASTTRACK" PROGRAM FOR ANALYZING  
LARGE NUMBERS OF SECURITIES IN A RAPID,  
EFFICIENT MANNER TO PROVIDE OPTIMUM  
CORRELATION OF ASSET RETURN TO A TIME DEPENDENT  
FINANCIAL INDEX. SUBSTANTIAL COMPUTER  
MEMORY STORAGE REDUCTION IS ALSO ACHIEVED.



```

C UPDATED JULY 10, 1989

C *****
C COPYWRITE 1989 NATIONAL INVESTMENT SERVICES OF AMERICA
C *****
C PROGRAM FASTRACK
C *****
C ***** INCLUDE COMMON BLOCK FILE *****
C *****
C ***** INCLUDE 'COMMON.F' *****
C *****
C ***** WRITE COPYRIGHT TO SCREEN *****
C *****
C ***** WRITE(*,*) 'COPYRIGHT (C) NATIONAL INVESTMENT SERVICES OF AME *****
C *****
C ***** WRITE(*,*) '***** 1988 *****
C *****
C ***** WRITE(*,*) 'Is the year that this unpublished work was origin *****
C *****
C ***** WRITE(*,*) 'created, NATIONAL INVESTMENT SERVICES OF AMERICA *****
C *****
C ***** WRITE(*,*) '("NISA") owns all rights to this work and intends *****
C *****
C ***** WRITE(*,*) 'maintain this work confidential so as to maintain *****
C *****
C ***** WRITE(*,*) 'work as a trade secret. NISA may also seek to *****
C *****
C ***** WRITE(*,*) 'maintain this work as an unpublished copyright. *****
C *****
C ***** WRITE(*,*) 'In the event of an inadvertent or deliberate *****
C *****
C ***** WRITE(*,*) 'publication NISA intends to enforce its rights *****
C *****
C ***** WRITE(*,*) 'to this work under the copyright laws as a published *****
C *****
C ***** WRITE(*,*) 'works. Those having access to this work may not copy, *****
C *****
C ***** WRITE(*,*) 'use, or disclose the information in this work unless *****
C *****
C ***** WRITE(*,*) 'expressly authorized by NISA to do so.' *****
C *****
C ***** WRITE(*,*) '***** CONFIDENTIAL *****
C *****
C ***** WRITE(*,*) 'PROPERTY OF NATIONAL INVESTMENT SERVICES OF AMERICA' *****
C *****
C ***** WRITE(*,*) 'ALL RIGHTS RESERVED' *****
C *****
C ***** WRITE(*,101)'CALLING COPYRIGHT *****
C *****
C ***** CALL COPYRIGHT *****
C *****
C ***** WRITE(*,101)'CALLING READFLAG ...
C *****
C ***** WRITE(*,101)'CALLING READFLAG ...
C *****
C ***** CALL READFLAG *****
C *****
C ***** WRITE(*,101)'CALLING DEFAULT ...
C *****
C ***** SET UP DEFAULT PARAMETERS *****
C *****
C ***** WRITE(*,101)'CALLING DEFAULT ...

```



109

```

DO 40 I = 1, IBUY-1
DO 50 J = 1, IBUY
IF (PART(BUY(J)), LT, PART(BUY(1))) THEN
ITEMP = BUY(1)
BUY(1) = BUY(J)
BUY(J) = ITEMP
ENDIF
CONTINUE
*** SORT THE SELLS ***
DO 60 I = 1, ISELL-1
TEST = PART(SELL(1))
IF (PART(SELL(J)) GT PART(SELL(1))) THEN
ITEMP = SELL(1)
SELL(1) = SELL(J)
SELL(J) = ITEMP
ENDIF
CONTINUE
***** DELTA IS THE PERCENTAGE IMPROVEMENT IN VARIANCE *****
DELTA = 0.0
IF (VOLD.NE.0.0) THEN
DELTA = (VOLD - VNEW)/VOLD
ENDIF
DELTA = DELTA * 100.
IBUY = 0
ISELL = 0
TURNOV = 0.0
TOTBU = 0.0
TOTSEL = 0.0
NUMSTK = 0
DO 30 I = 1, NUM
IF (X(I).GT.0.0001) NUMSTK=NUMSTK+1
IF (X(I)-OLDWT(I)) GT 0.00001 THEN
BUY(1BUY) = 1
TOTBU=TOTBU+X(I)-OLDWT(I)
ELSEIF (X(I)-OLDWT(I)) LT -0.00001 THEN
TOTSEL=TOTSEL+X(I)-OLDWT(I)
ENDIF
TURNOV = TURNOV + ABS(X(I)-OLDWT(I))
CONTINUE
***** SORT BASED ON PARTIALS *****
***** SORT THE BUYS *****
WRITE(10,*)
WRITE(10,102)(' ',K - 1,78)
FORMAT(A28,F18.2)

```

110

```

WR TE(10,102)('...',K - 1.78)
WR TE(10,*)' , SHARES ** SPAN SIC SYMBOL',
WR TE(10,*)' , PRICE DAYS SECURITY'
WR TE(10,*)' , OLD NEW DELTA SENS CODE IDC TICK',
WR TE(10,*)' , (00) (S) VOL NAME'
WR TE(10,102)('...',K - 1.78)
TOTBUY = 0.0
DO 80 1 = 1,1BUY
XOLD = OLDINT(BUY(1))*100.
XNEW = X(BUY(1))*100.
DELTA = XNEW - XOLD
***** CALCULATE THE ROUND LOT PURCHASE *****
IF (PRICE(BUY(1)).GT.0) THEN
  SHARES = 0.01*DELTA*PORTVAL/PRICE(BUY(1))
  SHARES = REAL(INT(SHARES/100.))
  TOTBUY = TOTBUY + SHARES*PRICE(BUY(1))*100.0
ELSE
  SHARES = 0.0
ENDIF
** DVOL IS IN DAILY DOLLAR VOLUME (000'S)
IF (DVOL(BUY(1)).GT.0) THEN
  DAYSVOL = 100.*SHARES*PRICE(BUY(1))/DVOL(BUY(1))/1000.0
ELSE
  DAYSVOL = 99.9
ENDIF
IF (RHS(BUY(1))-X(BUY(1)) .LT. 0.00001) THEN
  AROUND = 'A'
ELSE
  AROUND = ' '
ENDIF
IF (DELTA .GT. 0.01) THEN
  WRITE(10,101) XOLD,XNEW,DELTA,PART(BUY(1))*1000.,AROUND,
  ISIC(BUY(1)),IDC(BUY(1)),SHARES,
  PRICE(BUY(1)),DAYSVOL,NAME(BUY(1))
END IF
CONTINUE
WRITE(10,*)('T27,A12,F14.0') TOTAL BUY $ ,TOTBUY
WRITE(10,*)
WRITE(10,*) *****
WRITE(10,*) WRITE THE SELLS
WRITE(10,*) *****

```

```

WR TE(10,102)('...',K - 1.78)
WR TE(10,*)' , SHARES ** SPAN SIC SYMBOL',
WR TE(10,*)' , PRICE DAYS SECURITY'
WR TE(10,*)' , OLD NEW DELTA SENS CODE IDC TICK',
WR TE(10,*)' , (00) (S) VOL NAME'
WR TE(10,102)('...',K - 1.78)
TOTSELL = 0.0
DO 90 1 = 1,1SELL
XOLD = OLDINT(SELL(1))*100.
XNEW = X(SELL(1))*100.
DELTA = XOLD - XNEW
***** CALCULATE THE ROUND LOT SALE *****
IF (PRICE(SELL(1)).GT.0) THEN
  SHARES = 0.01*DELTA*PORTVAL/PRICE(SELL(1))
  SHARES = REAL(INT(SHARES/100.))
  TOTSELL = TOTSELL + SHARES*PRICE(SELL(1))*100.0
ELSE
  SHARES = 0.0
ENDIF
** DVOL IS IN DAILY DOLLAR VOLUME (000'S)
IF (DVOL(SELL(1)).GT.0) THEN
  DAYSVOL = 100.*SHARES*PRICE(SELL(1))/DVOL(SELL(1))/1000.0
ELSE
  DAYSVOL = 99.9
ENDIF
IF (RHS(SELL(1))-X(SELL(1)) .LT. 0.00001) THEN
  AROUND = 'A'
ELSE
  AROUND = ' '
ENDIF
IF (DELTA .GT. 0.01) THEN
  WRITE(10,101) XOLD,XNEW,DELTA,PART(SELL(1))*1000.,AROUND,
  ISIC(SELL(1)),IDC(SELL(1)),SHARES,
  PRICE(SELL(1)),DAYSVOL,NAME(SELL(1))
END IF
CONTINUE
IF (NONE.GT.0) THEN
  WRITE(10,*)
  WRITE(10,*) LISTING OF THE FORCED SELLS (STKS NOT AVAILABLE)
  WRITE(10,*) WEIGHT SOLD IDC SYMBOL
  WRITE(10,*)

```

```

IF(NMSAVE.GE.8701)CAP-CAPS* 0.769173
IF(NMSAVE.GE.8801)CAP-CAPS* 0.846097
IF(NMSAVE.GE.8901)CAP-CAPS* 1.000000

C WRITE(10,*)'IN BOUNDS AND MKTCAP = ',MKTCAP
IF (INSTOCKS.GT.0) THEN
  OPEN (9,FILE = STOKFIL,FORM='UNFORMATTED',STATUS = 'OLD')
  DO 600 I = 1,100000
    READ (9) (GARB,L = 1,3)
    READ (9) (GARB1, IGARB2, IGARB3, IGARB4
    READ (9) (XGARB1, XGARB2, XGARB3, XGARB4
    READ (9) (XGARB1, XGARB2, XGARB3, XGARB4
    READ (9) (XGARB1, XGARB2, XGARB3, XGARB4
    READ (9) (XGARB1, XGARB2, XGARB3, XGARB4
    DO 610 K = 1,100000-1
      READ (9) XGARB
      CONTINUE
    CONTINUE
  CONTINUE
  600
  610
  C
  620
  C
  630
  C
  640
  C
  650
  C
  660
  C
  670
  C
  680
  C
  690
  C
  700
  C
  710
  C
  720
  C
  730
  C
  740
  C
  750
  C
  760
  C
  770
  C
  780
  C
  790
  C
  800
  C
  810
  C
  820
  C
  830
  C
  840
  C
  850
  C
  860
  C
  870
  C
  880
  C
  890
  C
  900
  C
  910
  C
  920
  C
  930
  C
  940
  C
  950
  C
  960
  C
  970
  C
  980
  C
  990
  C
  1000
  C
  1010
  C
  1020
  C
  1030
  C
  1040
  C
  1050
  C
  1060
  C
  1070
  C
  1080
  C
  1090
  C
  1100
  C
  1110
  C
  1120
  C
  1130
  C
  1140
  C
  1150
  C
  1160
  C
  1170
  C
  1180
  C
  1190
  C
  1200
  C
  1210
  C
  1220
  C
  1230
  C
  1240
  C
  1250
  C
  1260
  C
  1270
  C
  1280
  C
  1290
  C
  1300
  C
  1310
  C
  1320
  C
  1330
  C
  1340
  C
  1350
  C
  1360
  C
  1370
  C
  1380
  C
  1390
  C
  1400
  C
  1410
  C
  1420
  C
  1430
  C
  1440
  C
  1450
  C
  1460
  C
  1470
  C
  1480
  C
  1490
  C
  1500
  C
  1510
  C
  1520
  C
  1530
  C
  1540
  C
  1550
  C
  1560
  C
  1570
  C
  1580
  C
  1590
  C
  1600
  C
  1610
  C
  1620
  C
  1630
  C
  1640
  C
  1650
  C
  1660
  C
  1670
  C
  1680
  C
  1690
  C
  1700
  C
  1710
  C
  1720
  C
  1730
  C
  1740
  C
  1750
  C
  1760
  C
  1770
  C
  1780
  C
  1790
  C
  1800
  C
  1810
  C
  1820
  C
  1830
  C
  1840
  C
  1850
  C
  1860
  C
  1870
  C
  1880
  C
  1890
  C
  1900
  C
  1910
  C
  1920
  C
  1930
  C
  1940
  C
  1950
  C
  1960
  C
  1970
  C
  1980
  C
  1990
  C
  2000
  C
  2010
  C
  2020
  C
  2030
  C
  2040
  C
  2050
  C
  2060
  C
  2070
  C
  2080
  C
  2090
  C
  2100
  C
  2110
  C
  2120
  C
  2130
  C
  2140
  C
  2150
  C
  2160
  C
  2170
  C
  2180
  C
  2190
  C
  2200
  C
  2210
  C
  2220
  C
  2230
  C
  2240
  C
  2250
  C
  2260
  C
  2270
  C
  2280
  C
  2290
  C
  2300
  C
  2310
  C
  2320
  C
  2330
  C
  2340
  C
  2350
  C
  2360
  C
  2370
  C
  2380
  C
  2390
  C
  2400
  C
  2410
  C
  2420
  C
  2430
  C
  2440
  C
  2450
  C
  2460
  C
  2470
  C
  2480
  C
  2490
  C
  2500
  C
  2510
  C
  2520
  C
  2530
  C
  2540
  C
  2550
  C
  2560
  C
  2570
  C
  2580
  C
  2590
  C
  2600
  C
  2610
  C
  2620
  C
  2630
  C
  2640
  C
  2650
  C
  2660
  C
  2670
  C
  2680
  C
  2690
  C
  2700
  C
  2710
  C
  2720
  C
  2730
  C
  2740
  C
  2750
  C
  2760
  C
  2770
  C
  2780
  C
  2790
  C
  2800
  C
  2810
  C
  2820
  C
  2830
  C
  2840
  C
  2850
  C
  2860
  C
  2870
  C
  2880
  C
  2890
  C
  2900
  C
  2910
  C
  2920
  C
  2930
  C
  2940
  C
  2950
  C
  2960
  C
  2970
  C
  2980
  C
  2990
  C
  3000
  C
  3010
  C
  3020
  C
  3030
  C
  3040
  C
  3050
  C
  3060
  C
  3070
  C
  3080
  C
  3090
  C
  3100
  C
  3110
  C
  3120
  C
  3130
  C
  3140
  C
  3150
  C
  3160
  C
  3170
  C
  3180
  C
  3190
  C
  3200
  C
  3210
  C
  3220
  C
  3230
  C
  3240
  C
  3250
  C
  3260
  C
  3270
  C
  3280
  C
  3290
  C
  3300
  C
  3310
  C
  3320
  C
  3330
  C
  3340
  C
  3350
  C
  3360
  C
  3370
  C
  3380
  C
  3390
  C
  3400
  C
  3410
  C
  3420
  C
  3430
  C
  3440
  C
  3450
  C
  3460
  C
  3470
  C
  3480
  C
  3490
  C
  3500
  C
  3510
  C
  3520
  C
  3530
  C
  3540
  C
  3550
  C
  3560
  C
  3570
  C
  3580
  C
  3590
  C
  3600
  C
  3610
  C
  3620
  C
  3630
  C
  3640
  C
  3650
  C
  3660
  C
  3670
  C
  3680
  C
  3690
  C
  3700
  C
  3710
  C
  3720
  C
  3730
  C
  3740
  C
  3750
  C
  3760
  C
  3770
  C
  3780
  C
  3790
  C
  3800
  C
  3810
  C
  3820
  C
  3830
  C
  3840
  C
  3850
  C
  3860
  C
  3870
  C
  3880
  C
  3890
  C
  3900
  C
  3910
  C
  3920
  C
  3930
  C
  3940
  C
  3950
  C
  3960
  C
  3970
  C
  3980
  C
  3990
  C
  4000
  C
  4010
  C
  4020
  C
  4030
  C
  4040
  C
  4050
  C
  4060
  C
  4070
  C
  4080
  C
  4090
  C
  4100
  C
  4110
  C
  4120
  C
  4130
  C
  4140
  C
  4150
  C
  4160
  C
  4170
  C
  4180
  C
  4190
  C
  4200
  C
  4210
  C
  4220
  C
  4230
  C
  4240
  C
  4250
  C
  4260
  C
  4270
  C
  4280
  C
  4290
  C
  4300
  C
  4310
  C
  4320
  C
  4330
  C
  4340
  C
  4350
  C
  4360
  C
  4370
  C
  4380
  C
  4390
  C
  4400
  C
  4410
  C
  4420
  C
  4430
  C
  4440
  C
  4450
  C
  4460
  C
  4470
  C
  4480
  C
  4490
  C
  4500
  C
  4510
  C
  4520
  C
  4530
  C
  4540
  C
  4550
  C
  4560
  C
  4570
  C
  4580
  C
  4590
  C
  4600
  C
  4610
  C
  4620
  C
  4630
  C
  4640
  C
  4650
  C
  4660
  C
  4670
  C
  4680
  C
  4690
  C
  4700
  C
  4710
  C
  4720
  C
  4730
  C
  4740
  C
  4750
  C
  4760
  C
  4770
  C
  4780
  C
  4790
  C
  4800
  C
  4810
  C
  4820
  C
  4830
  C
  4840
  C
  4850
  C
  4860
  C
  4870
  C
  4880
  C
  4890
  C
  4900
  C
  4910
  C
  4920
  C
  4930
  C
  4940
  C
  4950
  C
  4960
  C
  4970
  C
  4980
  C
  4990
  C
  5000
  C
  5010
  C
  5020
  C
  5030
  C
  5040
  C
  5050
  C
  5060
  C
  5070
  C
  5080
  C
  5090
  C
  5100
  C
  5110
  C
  5120
  C
  5130
  C
  5140
  C
  5150
  C
  5160
  C
  5170
  C
  5180
  C
  5190
  C
  5200
  C
  5210
  C
  5220
  C
  5230
  C
  5240
  C
  5250
  C
  5260
  C
  5270
  C
  5280
  C
  5290
  C
  5300
  C
  5310
  C
  5320
  C
  5330
  C
  5340
  C
  5350
  C
  5360
  C
  5370
  C
  5380
  C
  5390
  C
  5400
  C
  5410
  C
  5420
  C
  5430
  C
  5440
  C
  5450
  C
  5460
  C
  5470
  C
  5480
  C
  5490
  C
  5500
  C
  5510
  C
  5520
  C
  5530
  C
  5540
  C
  5550
  C
  5560
  C
  5570
  C
  5580
  C
  5590
  C
  5600
  C
  5610
  C
  5620
  C
  5630
  C
  5640
  C
  5650
  C
  5660
  C
  5670
  C
  5680
  C
  5690
  C
  5700
  C
  5710
  C
  5720
  C
  5730
  C
  5740
  C
  5750
  C
  5760
  C
  5770
  C
  5780
  C
  5790
  C
  5800
  C
  5810
  C
  5820
  C
  5830
  C
  5840
  C
  5850
  C
  5860
  C
  5870
  C
  5880
  C
  5890
  C
  5900
  C
  5910
  C
  5920
  C
  5930
  C
  5940
  C
  5950
  C
  5960
  C
  5970
  C
  5980
  C
  5990
  C
  6000
  C
  6010
  C
  6020
  C
  6030
  C
  6040
  C
  6050
  C
  6060
  C
  6070
  C
  6080
  C
  6090
  C
  6100
  C
  6110
  C
  6120
  C
  6130
  C
  6140
  C
  6150
  C
  6160
  C
  6170
  C
  6180
  C
  6190
  C
  6200
  C
  6210
  C
  6220
  C
  6230
  C
  6240
  C
  6250
  C
  6260
  C
  6270
  C
  6280
  C
  6290
  C
  6300
  C
  6310
  C
  6320
  C
  6330
  C
  6340
  C
  6350
  C
  6360
  C
  6370
  C
  6380
  C
  6390
  C
  6400
  C
  6410
  C
  6420
  C
  6430
  C
  6440
  C
  6450
  C
  6460
  C
  6470
  C
  6480
  C
  6490
  C
  6500
  C
  6510
  C
  6520
  C
  6530
  C
  6540
  C
  6550
  C
  6560
  C
  6570
  C
  6580
  C
  6590
  C
  6600
  C
  6610
  C
  6620
  C
  6630
  C
  6640
  C
  6650
  C
  6660
  C
  6670
  C
  6680
  C
  6690
  C
  6700
  C
  6710
  C
  6720
  C
  6730
  C
  6740
  C
  6750
  C
  6760
  C
  6770
  C
  6780
  C
  6790
  C
  6800
  C
  6810
  C
  6820
  C
  6830
  C
  6840
  C
  6850
  C
  6860
  C
  6870
  C
  6880
  C
  6890
  C
  6900
  C
  6910
  C
  6920
  C
  6930
  C
  6940
  C
  6950
  C
  6960
  C
  6970
  C
  6980
  C
  6990
  C
  7000
  C
  7010
  C
  7020
  C
  7030
  C
  7040
  C
  7050
  C
  7060
  C
  7070
  C
  7080
  C
  7090
  C
  7100
  C
  7110
  C
  7120
  C
  7130
  C
  7140
  C
  7150
  C
  7160
  C
  7170
  C
  7180
  C
  7190
  C
  7200
  C
  7210
  C
  7220
  C
  7230
  C
  7240
  C
  7250
  C
  7260
  C
  7270
  C
  7280
  C
  7290
  C
  7300
  C
  7310
  C
  7320
  C
  7330
  C
  7340
  C
  7350
  C
  7360
  C
  7370
  C
  7380
  C
  7390
  C
  7400
  C
  7410
  C
  7420
  C
  7430
  C
  7440
  C
  7450
  C
  7460
  C
  7470
  C
  7480
  C
  7490
  C
  7500
  C
  7510
  C
  7520
  C
  7530
  C
  7540
  C
  7550
  C
  7560
  C
  7570
  C
  7580
  C
  7590
  C
  7600
  C
  7610
  C
  7620
  C
  7630
  C
  7640
  C
  7650
  C
  7660
  C
  7670
  C
  7680
  C
  7690
  C
  7700
  C
  7710
  C
  7720
  C
  7730
  C
  7740
  C
  7750
  C
  7760
  C
  7770
  C
  7780
  C
  7790
  C
  7800
  C
  7810
  C
  7820
  C
  7830
  C
  7840
  C
  7850
  C
  7860
  C
  7870
  C
  7880
  C
  7890
  C
  7900
  C
  7910
  C
  7920
  C
  7930
  C
  7940
  C
  7950
  C
  7960
  C
  7970
  C
  7980
  C
  7990
  C
  8000
  C
  8010
  C
  8020
  C
  8030
  C
  8040
  C
  8050
  C
  8060
  C
  8070
  C
  8080
  C
  8090
  C
  8100
  C
  8110
  C
  8120
  C
  8130
  C
  8140
  C
  8150
  C
  8160
  C
  8170
  C
  8180
  C
  8190
  C
  8200
  C
  8210
  C
  8220
  C
  8230
  C
  8240
  C
  8250
  C
  8260
  C
  8270
  C
  8280
  C
  8290
  C
  8300
  C
  8310
  C
  8320
  C
  8330
  C
  8340
  C
  8350
  C
  8360
  C
  8370
  C
  8380
  C
  8390
  C
  8400
  C
  8410
  C
  8420
  C
  8430
  C
  8440
  C
  8450
  C
  8460
  C
  8470
  C
  8480
  C
  8490
  C
  8500
  C
  8510
  C
  8520
  C
  8530
  C
  8540
  C
  8550
  C
  8560
  C
  8570
  C
  8580
  C
  8590
  C
  8600
  C
  8610
  C
  8620
  C
  8630
  C
  8640
  C
  8650
  C
  8660
  C
  8670
  C
  8680
  C
  8690
  C
  8700
  C
  8710
  C
  8720
  C
  8730
  C
  8740
  C
  8750
  C
  8760
  C
  8770
  C
  8780
  C
  8790
  C
  8800
  C
  8810
  C
  8820
  C
  8830
  C
  8840
  C
  8850
  C
  8860
  C
  8870
  C
  8880
  C
  8890
  C
  8900
  C
  8910
  C
  8920
  C
  8930
  C
  8940
  C
  8950
  C
  8960
  C
  8970
  C
  8980
  C
  8990
  C
  9000
  C
  9010
  C
  9020
  C
  9030
  C
  9040
  C
  9050
  C
  9060
  C
  9070
  C
  9080
  C
  9090
  C
  9100
  C
  9110
  C
  9120
  C
  9130
  C
  9140
  C
  9150
  C
  9160
  C
  9170
  C
  9180
  C
  9190
  C
  9200
  C
  9210
  C
  9220
  C
  9230
  C
  9240
  C
  9250
  C
  9260
  C
  9270
  C
  9280
  C
  9290
  C
  9300
  C
  9310
  C
  9320
  C
  9330
  C
  9340
  C
  9350
  C
  9360
  C
  9370
  C
  9380
  C
  9390
  C
  9400
  C
  9410
  C
  9420
  C
  9430
  C
  9440
  C
  9450
  C
  9460
  C
  9470
  C
  9480
  C
  9490
  C
  9500
  C
  9510
  C
  9520
  C
  9530
  C
  9540
  C
  9550
  C
  9560
  C
  9570
  C
  9580
  C
  9590
  C
  9600
  C
  9610
  C
  9620
  C
  9630
  C
  9640
  C
  9650
  C
  9660
  C
  9670
  C
  9680
  C
  9690
  C
  9700
  C
  9710
  C
  9720
  C
  9730
  C
  9740
  C
  9750
  C
  9760
  C
  9770
  C
  9780
  C
  9790
  C
  9800
  C
  9810
  C
  9820
  C
  9830
  C
  9840
  C
  9850
  C
  9860
  C
  9870
  C
  9880
  C
  9890
  C
  9900
  C
  9910
  C
  9920
  C
  9930
  C
  9940
  C
  9950
  C
  9960
  C
  9970
  C
  9980
  C
  9990
  C
  10000
  C

```

```

491      IEN = 1
491      IF (GARB(ILEN:IEN:IEN).NE. ' ') THEN
491          IEN = IEN + 1
491      GOTO 491
491      ENDF
491      IDC(1)(1:IEN) = GARB(ILEN:ILEN:IEN-1)
491
491      ILEN = ILEN + IEN + 1
491      IF (GARB(ILEN:ILEN).EQ. ' ') THEN
491          ILEN = ILEN + 1
491          IF ((ILEN+3).GT.LENGTH) GOTO 793
491          GOTO 792
491      ENDF
491
491      IEN = 1
491      IF (GARB(ILEN:IEN:IEN).NE. ' ') THEN
491          IEN = IEN + 1
491          GOTO 492
491      ENDF
491      IDC(1)(6:6+IEN-1) = GARB(ILEN:ILEN:IEN-1)
491
491      CONTINUE
491
491      READ (9,END= 785) ISIC(1), IGARB1, IGARB2, IGARB3
491      IF (ISIC(1).GT.99) ISIC(1) = INT (ISIC(1)/100)
491      READ IN LINE 5 (PRICE,SPREAD)
491      READ(9,END = 785)PRICE(1),SPREAD(1),XGARB1,XGARB2
491      READ IN LINE 6 (MARKET CAP,VO
491      READ(9,END = 785)XMKTCAP(1),DVOL(1),XGARB1,XGARB2
491
491      IF (XMKTCAP(1).LT.CAP)INEXT=1
491
491      READ (9,END = 785) 1START
491      READ (9,END = 785) 1STOP
491
491      TEST FOR ENOUGH RETURNS TO SATISFY NRETS
491
491      ICHECK = NM-NRETS-1START
491      IF (ICHECK.LT.0) NRETS = NM - 1START
491      IF (NRETS.LT.NSTAT) ICHECK = 1FIRST-1START
491
491      DO 800 J = 1,ICHECK
491          READ (9) XGARB
491          CONTINUE
491
491      DO 801 J = 1,NRETS-NSTAT
491          READ(9) XGARB
491
491      800
491

```

```

      READ(13,'(A30)')DLOFILE
      CLOSE(13)
      RETURN
    ***** STOP DONE WITH MULTI RUN *****
    WRITE(*,*)'DONE WITH MULTIPLE RUN *****'
    !:TOP
101 FORMAT (A20)
      END
*****
***** SUBROUTINE MULTRUN *****
C ***** THIS SUBROUTINE SETS UP VARIABLE VALUES *****
C SUBROUTINE CALCPARAM *****
C ***** INCLUDE 'COMMON.F' *****
      NH = NMSAVE
      NN = (INT((NW/100)-80)*12+(NM-INT(NW/100))*100)
      IRAD = NSTAT+NSIMS
      IF 'NSTOCKS.GT.O')THEN
        'BUPPER' = SHAX
        ENI' IF
      MUP = NSTOCKS+NBNDS
      IF (NUM.GT.NMAX)THEN
        WRITE(*,*)' WARNING NUM > .NMAX.' TOO BIG FOR PROGRAM'
        STOP
      ENDIF
      ICO = ID
      KE = KEQ
      KKK = KE
      IA = ND
      ICC = MD
      IH = N2D
      IFIRST = NM-NSTAT
      ILAST = NM+NSIMS-1
      BULL1 = BULLET
      TARG1 = TARGET
      TURN1 = TURN
      ISKIP = ISKIP+1
      CLOSE(13)
      OPEN(13,FILE='PAST.DAT',STATUS='OLD')
      READ(13,*)GARB

```

```

114

      ICHANGE = 0
      XBLOWER = 0.0
      XBUPPER = 3.0
      XMAX = XBUPPER

      PRTVALS = XCONS(1)
      PRTVALS IS PORTVAL IN 1989 DOLLARS
      IT IS SCALED IN SUB. BOUNDS

      RETURN
      END

      ***** END OF SUBROUTINE DEFAULT *****

      ***** THIS SUBROUTINE READS IN STOCK DATA *****
      SUBROUTINE STOCKIN
      *****
      IFIRST IS FIRST MONTH OF DATA NEEDED FOR MATRIX
      ILAST IS LAST MONTH OF SIMULATION
      INCLUDE 'COMMON.F'

      CAPS-XCONS(4)

      READ IN DUMMY STOCK DATA
      THIS IS TO READ PAST

      SET UP CAP TO BE REPRESENTATIVE OF TIME PERIOD
      THE SCALARS ARE REP OF CAPS HISTORIES

      IF (NMSAV.EQ.7401)CAP-CAPS*
      IF (NMSAV.EQ.7501)CAP-CAPS*
      IF (NMSAV.EQ.7601)CAP-CAPS*
      IF (NMSAV.EQ.7701)CAP-CAPS*
      IF (NMSAV.EQ.7801)CAP-CAPS*
      IF (NMSAV.EQ.7901)CAP-CAPS*
      IF (NMSAV.EQ.8001)CAP-CAPS*
      IF (NMSAV.EQ.8101)CAP-CAPS*
      IF (NMSAV.EQ.8201)CAP-CAPS*
      IF (NMSAV.EQ.8301)CAP-CAPS*
      IF (NMSAV.EQ.8401)CAP-CAPS*
      IF (NMSAV.EQ.8501)CAP-CAPS*
      IF (NMSAV.EQ.8601)CAP-CAPS*
      IF (NMSAV.EQ.8701)CAP-CAPS*
      IF (NMSAV.EQ.8801)CAP-CAPS*
      IF (NMSAV.EQ.8901)CAP-CAPS*
      IF (CAP.EI.0)CAP-200000000.0

      ***** END OF SUBROUTINE CALCPARAM *****
      ***** THIS SUBROUTINE READS IN FLAG FILE *****
      SUBROUTINE READFLAG
      *****
      INCLUDE 'COMMON.F'

      OPEN(9,FILE='FLAG.NIS')
      DO 100 I=1,10
      READ(9,*)IFLAG(I)
      100 CONTINUE
      DO 200 I=1,10
      READ(9,*)XCONS(I)
      200 CONTINUE
      DO 300 I=1,10
      READ(9,*)AFLAG(I)
      300 CONTINUE
      CLOSE(9)
      RETURN
      END

      ***** END OF SUBROUTINE READFLAG *****
      ***** THIS SUBROUTINE SETS UP DEFAULT PARAMETERS *****
      SUBROUTINE DEFAULT
      *****
      INCLUDE 'COMMON.F'
      *****
      ***** DEFAULT READ IN OF 24 MONTHS PRIOR - MSTAT, NRESTS FOR B( ) *****
      MSTAT = 24
      NRESTS = 24
      BULLET = 0.0
      TARGET = 1.
      TURN = 100.
      ITYPE = 1
  
```

```

      BULLET = BULLET*ABS(BULLET)/1000.
      IF (TURN.EQ.100.)THEN
        TURN = 0.0
      ELSE
        XCON = 0.10
        TURN = .0001 * (100./(TURN+.01))-100.0/(100.01)
        A TURN OF .2 YIELDS A TURN - .4
      END IF
      RETURN
      END
  
```



```

792      ILEN = ILEN + IEN + 1
      IF (GARB(ILEN:IEN).EQ.' ') THEN
        ILEN = ILEN + 1
        IF ((ILEN+3).GT.LENGTH) GOTO 793
      ENDIF
      IEN = 1
      IF (GARB(ILEN:IEN+IEN).NE.' ') THEN
        IEN = IEN + 1
      ENDIF
      IDC(1)(6:6+IEN-1) = GARB(ILEN:ILEN+IEN-1)
      CONTINUE
      READ IN STOCK DATA
      NSTK=0
      DO 700 I = 1,NSTOCKS
        INEXT=0
        READ (9,111,END = 785) NAME(1)
        ILEN = 1
        GARB = 1
        IF (NAME(1)(ILEN:IEN).EQ.' ') THEN
          ILEN = ILEN+1
          GO TO 790
        ELSE
          GARB(1:LENGTH) = NAME(1)(ILEN:LENGTH)
          END IF
          NAME(1) = GARB
          WRITE (*,112) 1,NAME(1)
          READ (9,111,END = 785) GARB
          READ (9,111,END = 785) GARB
          IDC(1) = 1
          ILEN = 1
          IF (GARB(ILEN:IEN).EQ.' ') THEN
            ILEN = ILEN + 1
            IF ((ILEN+3).GT.LENGTH) GOTO 793
          ENDIF
          IEN = 1
          IF (GARB(ILEN:IEN+IEN).NE.' ') THEN
            IEN = IEN + 1
            GOTO 491
          ENDIF
          IDC(1)(1:IEN) = GARB(ILEN:ILEN+IEN-1)
        793
      790
      791
      792
      793
      794
      795
      796
      797
      798
      799
      800
      801
      802
      803
      804
      805
      806
      807
      808
      809
      810
      811
      812
      813
      814
      815
      816
      817
      818
      819
      820
      821
      822
      823
      824
      825
      826
      827
      828
      829
      830
      831
      832
      833
      834
      835
      836
      837
      838
      839
      840
      841
      842
      843
      844
      845
      846
      847
      848
      849
      850
      851
      852
      853
      854
      855
      856
      857
      858
      859
      860
      861
      862
      863
      864
      865
      866
      867
      868
      869
      870
      871
      872
      873
      874
      875
      876
      877
      878
      879
      880
      881
      882
      883
      884
      885
      886
      887
      888
      889
      890
      891
      892
      893
      894
      895
      896
      897
      898
      899
      900
      901
      902
      903
      904
      905
      906
      907
      908
      909
      910
      911
      912
      913
      914
      915
      916
      917
      918
      919
      920
      921
      922
      923
      924
      925
      926
      927
      928
      929
      930
      931
      932
      933
      934
      935
      936
      937
      938
      939
      940
      941
      942
      943
      944
      945
      946
      947
      948
      949
      950
      951
      952
      953
      954
      955
      956
      957
      958
      959
      960
      961
      962
      963
      964
      965
      966
      967
      968
      969
      970
      971
      972
      973
      974
      975
      976
      977
      978
      979
      980
      981
      982
      983
      984
      985
      986
      987
      988
      989
      990
      991
      992
      993
      994
      995
      996
      997
      998
      999

```

115

\*\*\* IF STOCK NON GOES AWAY DURING SIM PERIOD  
 \*\*\* THEN THE RETURNS TO MONEY MARKET TYPE LEVEL

116

```

IF (NBONDS.GT.0) THEN
  OPEN (9,FILE = BONDFIL, STATUS = 'OLD')
  DO 900 I = NSTOCKS+1,NSTOCKS+NBONDS
    READ (9,111) NAME(I)
    WRITE (*,113) I,NAME(I)
    READ (9,111) (GARB,L = 1,5)
    READ (9,*) ISTART
    READ (9,*) ISTOP
    ***** TEST ICHECK FOR ENOUGH RETURNS FOR BRET ***
    ICHECK = NM-NRETS-ISTART
    IF (ICHECK.LT.0) THEN
      WRITE(*,*) BOND THEN '1,' DOES NOT CONTAIN ENOUGH RETS
      FOR NRETS
      STOP
    END IF
    IF (NRETS.LT.NSTAT) ICHECK = IFIRST-ISTART
    DO 1000 J = 1,ICHECK
      READ (9,111) GARB
      CONTINUE
    DO 1001 J = 1,NRETS-NSTAT
      READ(9,*)GARB
      CONTINUE
    JJ = NRETS-NSTAT
    DO 1010 K = 1,JREAD
      READ (9,*) RET(K,1)
      J = JJ + K
      CONTINUE
    DO 1300 J = 1,ISTOP-ILAST
      READ (9,111) GARB
      CONTINUE
    900 CONTINUE
  ENDIF
  CLOSE(9)
  RETURN
  FORMAT (A40)
  FORMAT( '+','BOND # '15,' IS 'A40)
  END
  ***** END OF SUBROUTINE BONDIN *****
  C ***** THIS SUBROUTINE READS IN BOND DATA *****
  C ***** SUBROUTINE BONDIN *****
  C ***** IFIRST IS FIRST MONTH OF DATA NEEDED FOR MATRIX
  C ***** ILAST IS LAST MONTH OF SIMULATION *****
  C ***** INCLUDE 'COMMON.F' *****
  C ***** READ IN BOND DATA *****
  IF (K.GT.NSTAT.AND.RET(K,1).LT.-1.0) THEN
    RET(K,1)=-0.005
  ENDIF
  CONTINUE
  DO 1200 J = 1,ISTOP-ILAST
    READ (9,111) GARB
    CONTINUE
    IF (INEXT.EQ.1)GOTO 783
    NSTK=NSTK+1
    CONTINUE
  NSTOCKS = NSTK
  NUM = NSTOCKS+NBONDS
  ENDIF
  CLOSE(9)
  RETURN
  ***** IF END OF FILE WAS HIT ON STOCKS - RESET NUM AND CONTINUE ***
  C *****
  785 NSTOCKS = 1,1
  NUM = NSTOCKS+NBONDS
  CLOSE(9)
  RETURN
  CLOSE(9)
  FORMAT (A40)
  111 FORMAT( '+','STOCK # '15,' IS 'A40)
  112
  END
  ***** END OF SUBROUTINE STOCKIN *****
  C ***** THIS SUBROUTINE READS IN BOND DATA *****
  C ***** SUBROUTINE BONDIN *****
  C ***** IFIRST IS FIRST MONTH OF DATA NEEDED FOR MATRIX
  C ***** ILAST IS LAST MONTH OF SIMULATION *****
  C ***** INCLUDE 'COMMON.F' *****
  C ***** READ IN BOND DATA *****

```

```

DO 900 J = 1,NUM
PRET(1) = PRET(1) + RET(1,J)*X(J)
CONTINUE
OBJ = OBJ + (PRET(1)-ANIM(1))**2/REAL(NSTAT)
AVEDIFF = AVEDIFF + (PRET(1)-ANIM(1))/REAL(NSTAT)
CONTINUE
IF (OBJ .GT. 0.0) THEN
OBJ = ((OBJ-AVEDIFF**2)*12.)**(0.5)*100
ELSE
OBJ = 99.9
ENDI
C
C NOW CALCULATE RETURNS AND WEIGHTS IN SIMULATION PERIOD *
DO 801 I=1,NSTATS
PRET(I+NSTAT)=0.0
DO 901 J=1,NUM
PRET(I+NSTAT) = PRET(I+NSTAT)+CUM(J)*RET(I+NSTAT,J)
CONTINUE
901 CONTINUE
30 1001 K = 1,NUM
CUM(K) = CUM(K)*(1.+RET(I+NSTAT,K))/(1.+PRET(I+NSTAT))
CONTINUE
1001 CONTINUE
801 CONTINUE
C
C WRITE OUT THE NEW WEIGHTS TO A FILE CALLED 'OLD'
C OPEN(15,FILE = 'OLD', STATUS = 'OLD')
C REWID(15)
C DO 140 I = 1,NUM
C WRIT(15,940) CUM(I)*100
C CONTINUE
C C140 FORMU(TIX,F15.8)
C C940 CLOSE(15)
C
C WRITE OUTPUT *****
OPEN(1),FILE=FILEOUT,STATUS='UNKNOWN')
REWIND(10)
WRITE(10,*) NATIONAL INVESTMENT SERVICES SPANNING TECHNOLOGY'
WRITE(10,*) SIMULATION RESULTS'
WRITE(10,*) ',TITLE'
WRITE(10,*) LIABILITY STREAM USED ',LIANAME'
WRITE(10,*) START END SPAN'
WRITE(10,*) NUMBER WGT WHT SENSIT SIC IDC TICK',
A WRITE(10,*) 'SECURITY NAME'
C
C CALCULATE THE PORTFOLIO RETURNS FIRST *****
OBJ = 0.0
AVEDIFF = 0.0
DO 800 I = 1,NSTAT
PRET(I) = 0.0

```

```

DO 250 I=NSTAT+1,IREAD
  AVE(1)=AVE(1)+ANIM(1)/REAL(NSIM)
  AVE(2)=AVE(2)+PRET(1)/REAL(NSIM)
  AVE(3)=AVE(3)+PRET(1)-ANIM(1)/REAL(NSIM)
260 CONTINUE
C *****
  CALCULATE COVARIANCES *****
DO 910 I=1,3
  VAR(1)=0.0
  DO 950 K=NSTAT+1,IREAD
    VAR(1)=VAR(1)+((STAT(K,1)-AVE(1))**2)/REAL(NSIM-1)
950 CONTINUE
C *****
  CALCULATE THE COVARIANCE MATRIX'S UPPER OFF DIAGONAL *****
DO 960 I=1,3
  DO 960 J=1,3
    DO 828 K=NSTAT+1,IREAD
      CHAT(1,J)=CHAT(1,J)+((STAT(K,1)-AVE(1))
        * (STAT(K,J)-AVE(J)))
      CONTINUE
960 CONTINUE
      CHAT(J,1) = CHAT(1,J)
C *****
  CALCULATE CORRELATIONS *****
DO 951 I=1,3
  DO 951 J=1,1
    IF (VAR(1).LE. 0.0 .OR. VAR(J).LE. 0.0) GO TO 951
    C(REL(1,J))=CHAT(1,J)/(VAR(1)**.5*VAR(J)**.5)
951 CONTINUE
  CALCULATE CUMULATIVE RETURNS *****
PRET(NSTAT)=100
ANIM(NSTAT)=100
DO 569 I=NSTAT+1,IREAD
  XX=RET(1)
  YY=WMIM(1)
  PRET(1)=PRET(1-1)*(1+XX)
  ANIM(1)=ANIM(1-1)*(1+YY)
569 CONTINUE
  IF (VAR(1).NE. 0.0) BETA = CHAT(1,2)/VAR(1)
  ALPHA = AVE(2) - BETA*AVE(1)
  SST = 0.0
  SSE = 0.0
DO 952 I = NSTAT+1, IREAD
  SSE = SSE + (ALPHA+BETA*STAT(1,1)-STAT(1,2))**2
  SST = SST + (STAT(1,2)-AVE(2))**2

```

```

XRET=0.0
DO 131 I=1,NUM
  IF (IFLAG(9).EQ.2.AND.X(1).LT.
    WRITE(10,16) I,X(1)-100,COR(1)-100,PART(1)-100,
    ISIC(1),IDC(1),NAME(1)
631
  A
  XRETI = AVE(1) + UANIM
  XRET = XRET + X(1)*AVE(1)
  CONTINUE
131
16
  FORMAT(1X,14,1X,2F6.3,F7.3,14,1X,A9,1X,A36)
  XRET = XRET*12.0*100.0
  WRITE(10,*)
  WRITE(10,849) ORJ, XRET
  FORMAT(1X, MINIMUM STD DEV =',F10.4,' EXPECTED RETURN =',F10.4)
649
  RETURN
END
C *****
  END OF SUBROUTINE PORT *****
C *****
  SUBROUTINE TRACKER *****
C *****
  INCLUDE 'COMMON.F' *****
C *****
  SET AVERAGES AND COVARIANCES TO ZERO *****
DO 400 I=1,3
  AVE(I)=0.0
  DO 400 J=1,3
    CHAT(I,J)=0.0
400 CONTINUE
C ***
  SET STAT ARRAY 1 = LIABILITY, 2 = PORTFOLIO, 3 = DIFFERENCES **
DO 250 I=1,IREAD
  STAT(1,1)=ANIM(1)
  STAT(1,2)=PRET(1)
  STAT(1,3)=PRET(1)-ANIM(1)
250 CONTINUE
C *****
  GO AROUND STATISTICS IF NSIM < 3
  IF (NSIMS.LT.3) GO TO 318
  *****
  CALCULATE AVERAGES *****
  *****
  SET NSIM TO 2 FOR STATISTICS IF NSIMS.LT.1 *****
  NSIM = NSIMS
  IF (NSIMS.LT.1) NSIM = 2

```

119

```

988 FORMAT(1X,A12.10X,F10.6,X,F10.2,' %')
989 FORMAT(1X,A12.10X,F10.6)

C ***** WRITE AVERAGES AND VARIANCES OF 1-LIAB, 2-PORT, 3-DIFFS **
WRITE(10,*)
WRITE(10,*)
WRITE(10,*) ANNUAL RETURNS:
WRITE(10,*)
DO 990 I = 1,3
  AVE(I) = (AVE(I)*12)+100
  VAR(I) = (VAR(I)*12)**.5*.100
CONTINUE
990

WRITE(10,916) 'LIABILITY', AVE(1), VAR(1)
WRITE(10,916) 'PORTFOLIO', AVE(2), VAR(2)
WRITE(10,916) 'DIFFERENCE', AVE(3), VAR(3)
916 FORMAT(1X,3X,A11.9X,F9.2,' %',10X,F9.2,' %')

C ***** WRITE PARAMETERS FOR THE RUN *****
WRITE(10,*)
WRITE(10,*)
WRITE(10,*)
WRITE(10,*)
WRITE(10,101) TITLE
WRITE(10,102) FILEOUT
WRITE(10,103) LIABFIL
WRITE(10,104) STOKFIL
WRITE(10,105) BONDFIL
WRITE(10,106) NMSAVE
WRITE(10,107) NSIMS
WRITE(10,108) NSTOCKS
WRITE(10,109) NBONDS
WRITE(10,110) IDUMMY
WRITE(10,111) STOCKMIN
WRITE(10,112) YIPAK
WRITE(10,113) SHAX
WRITE(10,114) NSTAT
WRITE(10,115) NRETS
WRITE(10,116) BULL1
WRITE(10,117) TARG1
WRITE(10,118) TURN1

FORNAT(1X, 1) SPANNING RUN TITLE
FORNAT(1X, 2) OUTPUT FILENAME
FORNAT(1X, 3) LIABILITY STREAM FILE
FORNAT(1X, 4) STOCK RETURN FILE
FORNAT(1X, 5) BOND RETURN FILE
FORNAT(1X, 6) MONTH SIMULATION BEGINS
FORNAT(1X, 7) NUMBER OF MONTHS SIMULATED

988
989
C *****
IF(NSIMS.GT.2) STDERR = (SSE/REAL(NSIMS-2))**.5
IF(SST.NE.0.0) RSQ = 1 - SSE/SST

C ***** BEGIN WRITING THE TRACKER OUTPUT *****
WRITE(10,*)
WRITE(10,*) TRACKER OUTPUT FILE
WRITE(10,*) MONTH LIA PORT DIF
WRITE(10,*)

318 WRITE(10,*)
WRITE(10,*)
WRITE(10,*)

C ***** WRITE MONTH, LIABILITY RET, PORTFOLIO RET, DIFFERENCE *****
DO 700 I=NSTAT+1,IREAD
  ISTAT = I-(NSTAT+1)
  WRITE(10,612) ISTAT+NM,(STAT(I,J),J=1,3)
  FORMAT(1X,3X,14.3F13.10)
612 CONTINUE
700 CONTINUE

C ***** WRITE CUMULATIVE RETURNS *****
WRITE(10,*)
WRITE(10,886)
FORMAT(1X,'CUMULATIVE VALUES OF LIABILITIES AND ASSETS',/
/, 'MONTH LIABILITIES PORTFOLIO',/)

886
DO 847 I=NSTAT,IREAD
  ISTAT = I-(NSTAT+1)
  WRITE(10,328) ISTAT+NM,ANIM(1),PRET(1)
  FORMAT(1X,15.7X,F7.3,10X,F7.3)
328 CONTINUE
847 CONTINUE
837 CONTINUE

C ***** WRITE STATISTICS BASED ON SIMULATIONS MONTHS *****
WRITE(10,*)
WRITE(10,*)
WRITE(10,985) NM,NSIMS-1
FORMAT(/, 'STATISTICS BASED ON MONTHS ',14, ' THROUGH ',14,

985
C ***** WRITE ALPHA,BETA,STDERR,CORREL,AND R-SQUARED *****
WRITE(10,*)
WRITE(10,987) 'ALPHA', 'ALPHA', 'ANNUAL', 'ALPHA*1200
WRITE(10,989) 'BETA', 'BETA
WRITE(10,988) 'STD ERR', 'STDERR*100
WRITE(10,989) 'CORRELATION', 'CORREL(2,1)
WRITE(10,988) 'R-SQUARED', 'RSQ, RSQ*100
FORMAT(1X,A12.10X,F10.6,AB,F10.2,' %')

987

```

```

SIC(25) - 'PAPER AND ALLIED PRODUCTS'
SIC(27) - 'PRINTING PUBLISHING AND A.P.'
SIC(28) - 'CHEMICAL AND ALLIED PROD.'
SIC(29) - 'PETROLEUM REFIN. & REL. PROD.'
SIC(30) - 'RUBBER & LEATHER PRODUCTS'
SIC(31) - 'STONE CLAY GLASS & CONC. PROD.'
SIC(32) - 'PRIMARY METAL INDUSTRIES'
SIC(33) - 'MACHINERY EXCEPT ELECTRICAL'
SIC(34) - 'ELECTRIC MACH. & EQUIP.'
SIC(35) - 'ELECTRIC GAS AND SANIT. SERV.'
SIC(36) - 'WHOLESALE TRADE-DURABLE GOODS'
SIC(37) - 'WHOLESALE TRADE-NONDUR. GOODS'
SIC(38) - 'BUILDING MAT. WARE. GAR. SUPP.'
SIC(39) - 'GENERAL MERCH. STORES'
SIC(40) - 'FOOD STORES'
SIC(41) - 'AUTOMOTIVE DEAL AND GAS. SS'
SIC(42) - 'APPAREL AND ACCESS.'
SIC(43) - 'HOME FURN. AND EQUIP. STORES'
SIC(44) - 'EATING AND DRINKING PLACES'
SIC(45) - 'MISCELLANEOUS RETAIL'
SIC(46) - 'BANKING'
SIC(47) - 'CRED. AGEN. OTH. THAN BANKS'
SIC(48) - 'SEC. AND COMM. BROK. DEENSE'
SIC(49) - 'INSURANCE CARRIERS'
SIC(50) - 'INS. AGENTS BROK. SERV.'
SIC(51) - 'REAL ESTATE'
SIC(52) - 'CORP. RE. INS. LOANS & LAW OFF.'
SIC(53) - 'HOLD. AND OTHER INV. COMP.'
SIC(54) - 'HOTELS ROOM. HOUSES CAMP AOLP'
SIC(55) - 'PERSONAL SERVICES'
SIC(56) - 'BUSINESS SERVICES'
SIC(57) - 'AUTO. REPAIR SERV. AND GAR.'
SIC(58) - 'MISC. REPAIR SERVICES'
SIC(59) - 'MOTION PICTURES'
SIC(60) - 'AMUSE. AND REC. SERV. EX. MP'
SIC(61) - 'HEALTH SERVICES'
SIC(62) - 'LEGAL SERVICES'
SIC(63) - 'EDUCATIONAL SERVICES'

- 'PAPER AND ALLIED PRODUCTS'
- 'PRINTING PUBLISHING AND A.P.'
- 'CHEMICAL AND ALLIED PROD.'
- 'PETROLEUM REFIN. & REL. PROD.'
- 'RUBBER & LEATHER PRODUCTS'
- 'STONE CLAY GLASS & CONC. PROD.'
- 'PRIMARY METAL INDUSTRIES'
- 'MACHINERY EXCEPT ELECTRICAL'
- 'ELECTRIC MACH. & EQUIP.'
- 'ELECTRIC GAS AND SANIT. SERV.'
- 'WHOLESALE TRADE-DURABLE GOODS'
- 'WHOLESALE TRADE-NONDUR. GOODS'
- 'BUILDING MAT. WARE. GAR. SUPP.'
- 'GENERAL MERCH. STORES'
- 'FOOD STORES'
- 'AUTOMOTIVE DEAL AND GAS. SS'
- 'APPAREL AND ACCESS.'
- 'HOME FURN. AND EQUIP. STORES'
- 'EATING AND DRINKING PLACES'
- 'MISCELLANEOUS RETAIL'
- 'BANKING'
- 'CRED. AGEN. OTH. THAN BANKS'
- 'SEC. AND COMM. BROK. DEENSE'
- 'INSURANCE CARRIERS'
- 'INS. AGENTS BROK. SERV.'
- 'REAL ESTATE'
- 'CORP. RE. INS. LOANS & LAW OFF.'
- 'HOLD. AND OTHER INV. COMP.'
- 'HOTELS ROOM. HOUSES CAMP AOLP'
- 'PERSONAL SERVICES'
- 'BUSINESS SERVICES'
- 'AUTO. REPAIR SERV. AND GAR.'
- 'MISC. REPAIR SERVICES'
- 'MOTION PICTURES'
- 'AMUSE. AND REC. SERV. EX. MP'
- 'HEALTH SERVICES'
- 'LEGAL SERVICES'
- 'EDUCATIONAL SERVICES'

108 FORMAT(IX, 8) NUMBER OF STOCKS
109 FORMAT(IX, 9) NUMBER OF BONDS
110 FORMAT(IX, 10) STOCKS TO READ PAST
111 FORMAT(IX, 11) MINIMUM STOCK HOLDING
112 FORMAT(IX, 12) MAXIMUM STOCK HOLDING
113 FORMAT(IX, 13) % F6.2
114 FORMAT(IX, 14) % F6.2
115 FORMAT(IX, 15) NSTAT
116 FORMAT(IX, 16) NETS
117 FORMAT(IX, 17) BULLET
118 FORMAT(IX, 18) TARGET
119 FORMAT(IX, 19) FACTOR
120 FORMAT(IX, 20) TURNOVER

C ***** WRITE HISTORICAL DATA USED TO MAKE RUN *****
C WRITE(10,*) PORTFOLIO CONSTRUCTED UPON THE FOLLOWING DATA:
C WRITE(10,*) MONTH LIA PORT DIF
C WRITE(10,*)
C WRITE(10,*)
C DO 300 I=1,NSTAT
C CONTINUE
300 RETURN
END
C ***** END OF SUBROUTINE TRACKER *****
C ***** SUBROUTINE INDUST *****
C ***** INCLUDE 'COMMON.F' *****
C
SIC(1) - 'AGRICULTURAL PRODUCTION-CROPS'
SIC(2) - 'AGRICULTURAL PROD.-LIVESTOCK'
SIC(3) - 'AGRICULTURAL SERVICES'
SIC(4) - 'FORESTRY'
SIC(5) - 'FISHING HUNTING AND TRAPPING'
SIC(6) - 'METAL MINING'
SIC(7) - 'ANTHRACITE MINING'
SIC(8) - 'BITUMINOUS COAL & LIGNITE MIN.'
SIC(9) - 'OIL & GAS EXTRACTION'
SIC(10) - 'MIN. & QUARRY. OF NONMET. MIN.'
SIC(11) - 'BUILD. CONST.-G. C. & O. B.'
SIC(12) - 'CONST. OTH THAN B.C.-G.C.'
SIC(13) - 'CONST.-SPEC. TRADE CONT.'
SIC(14) - 'FOOD KINDRED PRODUCTS'
SIC(15) - 'TOBACCO MANUFACTURERS'
SIC(16) - 'TEXTILE MILL PRODUCTS'
SIC(17) - 'APP. & OTH. FIN. PROD. MFAOSH'
SIC(18) - 'LUMBER & WOOD PROD. EX. FURN.'
SIC(19) - 'FURNITURE AND FIXTURES'
SIC(20) - 'SIC(21)'
SIC(21) - 'SIC(22)'
SIC(22) - 'SIC(23)'
SIC(23) - 'SIC(24)'
SIC(24) - 'SIC(25)'
SIC(25) - 'SIC(26)'
SIC(26) - 'SIC(27)'
SIC(27) - 'SIC(28)'
SIC(28) - 'SIC(29)'
SIC(29) - 'SIC(30)'
SIC(30) - 'SIC(31)'
SIC(31) - 'SIC(32)'
SIC(32) - 'SIC(33)'
SIC(33) - 'SIC(34)'
SIC(34) - 'SIC(35)'
SIC(35) - 'SIC(36)'
SIC(36) - 'SIC(37)'
SIC(37) - 'SIC(38)'
SIC(38) - 'SIC(39)'
SIC(39) - 'SIC(40)'
SIC(40) - 'SIC(41)'
SIC(41) - 'SIC(42)'
SIC(42) - 'SIC(43)'
SIC(43) - 'SIC(44)'
SIC(44) - 'SIC(45)'
SIC(45) - 'SIC(46)'
SIC(46) - 'SIC(47)'
SIC(47) - 'SIC(48)'
SIC(48) - 'SIC(49)'
SIC(49) - 'SIC(50)'
SIC(50) - 'SIC(51)'
SIC(51) - 'SIC(52)'
SIC(52) - 'SIC(53)'
SIC(53) - 'SIC(54)'
SIC(54) - 'SIC(55)'
SIC(55) - 'SIC(56)'
SIC(56) - 'SIC(57)'
SIC(57) - 'SIC(58)'
SIC(58) - 'SIC(59)'
SIC(59) - 'SIC(60)'
SIC(60) - 'SIC(61)'
SIC(61) - 'SIC(62)'
SIC(62) - 'SIC(63)'
SIC(63) - 'SIC(64)'
SIC(64) - 'SIC(65)'
SIC(65) - 'SIC(66)'
SIC(66) - 'SIC(67)'
SIC(67) - 'SIC(68)'
SIC(68) - 'SIC(69)'
SIC(69) - 'SIC(70)'
SIC(70) - 'SIC(71)'
SIC(71) - 'SIC(72)'
SIC(72) - 'SIC(73)'
SIC(73) - 'SIC(74)'
SIC(74) - 'SIC(75)'
SIC(75) - 'SIC(76)'
SIC(76) - 'SIC(77)'
SIC(77) - 'SIC(78)'
SIC(78) - 'SIC(79)'
SIC(79) - 'SIC(80)'
SIC(80) - 'SIC(81)'
SIC(81) - 'SIC(82)'

```

INCLUDE 'COMMON.F'

OPEN (9,FILE - LIAFIL, STATUS - 'OLD')

TRITE(+'+',A50)-)LIANAME

READ (9,11) GARB

HEAD	{9,*}	ISTART
HEAD	{9,*}	ISTOP

0 300 ! - 1,IFIRST-ISTART  
READ (9,111) GARB  
N=1N1E

100 I = 1 IREAD

```

100 I = 1, IREAD
    READ (9,*) ANIM(I)
    CONTINUE

```

**(6):15073**

**RETURN**

111 FORM 1 (A40)

```

C ***** 1*
C      END OF SUBROUTINE
C      UPDATED 4/18/88
C ***** LIASUB *****

```

```
C*****  
C      7/10/88  
SUBROUTINE TO WRITE OUT *****
```

\*\*\*\*\* SUBROUTINE TO WRITE OUT COPYRIGHT WARNING \*\*\*\*\*

[illegible]

[illegible]



```

202 ELSE IF (IRESPON.EQ.2) THEN
    WRITE(*,122)
    READ(*, '(A30)', END = 202, ERR = 202) FILEOUT
203 ELSE IF (IRESPON.EQ.3) THEN
    WRITE(*,123)
    READ(*, '(A30)', END = 203, ERR = 203) LIAFIL
204 ELSE IF (IRESPON.EQ.4) THEN
    WRITE(*,124)
    READ(*, '(A30)', END = 204, ERR = 204) STOKFIL
205 ELSE IF (IRESPON.EQ.5) THEN
    WRITE(*,125)
    READ(*, '(A30)', END = 205, ERR = 205) BONOFIL
206 ELSE IF (IRESPON.EQ.6) THEN
    WRITE(*,126)
    READ(*, *, END = 206, ERR = 206) MMSAVE
207 ELSE IF (IRESPON.EQ.7) THEN
    WRITE(*,127)
    READ(*, *, END = 207, ERR = 207) MSHMS
208 ELSE IF (IRESPON.EQ.8) THEN
    WRITE(*,128)
    READ(*, *, END = 208, ERR = 208) NSTOCKS
209 ELSE IF (IRESPON.EQ.9) THEN
    WRITE(*,129)
    READ(*, *, END = 209, ERR = 209) NBONDS
210 ELSE IF (IRESPON.EQ.10) THEN
    WRITE(*,130)
    READ(*, *, END = 210, ERR = 210) IDUMMY
211 ELSE IF (IRESPON.EQ.11) THEN
    WRITE(*,131)
    READ(*, *, END = 211, ERR = 211) STOCKMIN
212 ELSE IF (IRESPON.EQ.12) THEN
    WRITE(*,132)
    READ(*, *, END = 212, ERR = 212) YIMAX
213 ELSE IF (IRESPON.EQ.13) THEN
    WRITE(*,133)
    READ(*, *, END = 213, ERR = 213) SNAX
214 ELSE IF (IRESPON.EQ.14) THEN
    WRITE(*,134)
    READ(*, *, END = 214, ERR = 214) NSTAT

```

10

```

1 101 WRITE(*,101)
    FORMAT(1) ENTER TITLE FOR SPANNING RUN
    READ(*,A30),END - 1,ERR - 1) TITLE
    WRITE(*,*)

2 102 WRITE(*,102)
    FORMAT(2) ENTER OUTPUT FILE NAME
    READ(*,A30),END - 2,ERR - 2) FILEOUT
    WRITE(*,*)

3 103 WRITE(*,103)
    FORMAT(3) ENTER LIABILITY RETURNS FILE NAME
    READ(*,A30),END - 3,ERR - 3) LIAFIL
    WRITE(*,*)

4 104 WRITE(*,104)
    FORMAT(4) ENTER STOCK RETURNS FILE NAME
    READ(*,A30),END - 4,ERR - 4) STOKFIL
    IF (STOKFIL.EQ.'') STOKFIL = 'STOCKS.PRN'
    WRITE(*,*)

5 105 WRITE(*,105)
    FORMAT(5) ENTER BOND RETURNS FILE NAME
    READ(*,A30),END - 5,ERR - 5) BONDFIL
    IF (BONDFIL.EQ.'') BONDFIL = 'BONDS.PRN'
    WRITE(*,*)

6 106 WRITE(*,106)
    FORMAT(6) ENTER YEAR AND MONTH TO BEGIN
    READ(*,*) SIMULATION (6,9, FEB 1987 - 8702)
    WRITE(*,*)

7 107 WRITE(*,107)
    FORMAT(7) ENTER NUMBER OF MONTHS TO SIMULATE
    READ(*,*) NSIMS
    WRITE(*,*)

8 108 WRITE(*,108)
    FORMAT(8) ENTER NUMBER OF STOCKS TO USE
    READ(*,*) NSTOCKS
    WRITE(*,*)

9 109 WRITE(*,109)
    FORMAT(9) ENTER NUMBER OF BONDS TO USE
    READ(*,*) NBONDS
    WRITE(*,*)

10 110 WRITE(*,110)
    FORMAT(10) ENTER NUMBER OF STOCKS READ PAST
    READ(*,*) IDUMMY
    WRITE(*,*)

20 CALL MENU (LINES,HEAD,LAST,NBROPT,OPTIONS,IRESPO)
    IF (IRESPO.EQ.1) THEN
        WRITE(*,100)(' ',1 - 1,5)
        WRITE(*,104)
        READ(*,A30),END - 20,ERR - 20) PAST
        GO TO 5
    ELSE IF (IRESPO.EQ.2) THEN
        CALL REDSCRN
        RETURN
    END IF
    RETURN

100 FORMAT('0',A1)
101 FORMAT(A35)
102 FORMAT(14)
103 FORMAT(F7.2)
104 FORMAT(' ', NAME FOR PAST DATA FILE --- ', $)
    END

***** END OF SUBROUTINE REDPAST *****
***** SUBROUTINE REDSCRN *****
C TITLE
C FILEOUT
C LIAFIL
C STOKFIL
C BONDFIL
C NSIMS
C NSTOCKS
C NBONDS
C IDUMMY
C STOCKMIN
C YIMAX
C
C TITLE FOR THIS RUN
C OUTPUT FILE
C LIABILITY RETURNS FILE
C STOCK RETURNS FILE
C BOND RETURNS FILE
C FIRST MONTH OF SIMULATION
C NUMBER OF MONTHS TO SIMULATE
C NUMBER OF STOCKS TO SEND TO OPTIMIZER
C NUMBER OF BONDS TO SEND TO OPTIMIZER
C NUMBER OF STOCKS TO READ PAST IN STOKFIL
C MINIMUM HOLDING IN STOCKS
C INDUSTRY MAXIMUM HOLDING
C
C INCLUDE 'COMMON.F'
C
C ** DEFAULT NAMES FOR STOCK AND BOND FILES
C ** STOCKS.PRN & BONDS.PRN
C
113 WRITE(*,113)(' ',1 - 1,12)
    FORMAT('0',A1)

```

126

```

101 FORMAT(IX, 1) SPANNING RUN TITLE
102 FORMAT(IX, 2) OUTPUT FILENAME
103 FORMAT(IX, 3) LIABILITY STREAM FILE
104 FORMAT(IX, 4) STOCK RETURN FILE
105 FORMAT(IX, 5) BOND RETURN FILE
106 FORMAT(IX, 6) MONTH OF MONTHS TO SIMULATE
107 FORMAT(IX, 7) NUMBER OF MONTHS TO SIMULATE
108 FORMAT(IX, 8) NUMBER OF BONDS
109 FORMAT(IX, 9) NUMBER OF STOCKS
110 FORMAT(IX, 10) STOCKS TO READ PAST
111 FORMAT(IX, 11) MINIMUM STOCK HOLDING
112 FORMAT(IX, 12) MAXIMUM STOCK HOLDING
113 FORMAT(IX, 13) MAXIMUM INDUSTRY HOLDING
114 FORMAT(IX, 14) NSTAT
115 FORMAT(IX, 15) NRETS
116 FORMAT(IX, 16) BULLET
117 FORMAT(IX, 17) TARGET
118 FORMAT(IX, 18) TURNOVER FACTOR
*****
***** END OF SUBROUTINE DISPLAY *****
*****
***** SUBROUTINE SAVDAT *****
*****
* ** THIS ROUTINE SAVE THE FILE FOR LATER USE
* **
* **
*****
***** INCLUDE 'COMMON.F' *****
C WRITE(*,100)(' ',1=1,7)
C WRITE(*,100)(' ',1=1,5)
*****
OPEN(8,FILE='PAST.DAT',STATUS='OLD')
WRITE(8,101) TITLE
WRITE(8,102) FILEOUT
WRITE(8,103) LIAFIL
WRITE(8,104) STOKFIL
WRITE(8,105) BONDFIL
WRITE(8,106) NSAVE
WRITE(8,107) NSIMS
WRITE(8,108) NSTOCKS
WRITE(8,109) NBONDS
WRITE(8,110) IDUMMY
WRITE(8,111) STOCKMIN
WRITE(8,112) YIMAX
WRITE(8,113) SHAX
WRITE(8,114) NSTAT
WRITE(8,115) NRETS
WRITE(8,116) BULLET
WRITE(8,117) TARGET
WRITE(8,118) TURN
CLOSE (8)
RETURN
100 FORMAT(' ',A1)

```

```

*****
***** SUBROUTINE DISPLAY *****
*****
* ** THIS ROUTINE DISPLAYS SELECTED PARAMETERS FOR SPANNING RUN
* **
* **
*****
***** INCLUDE 'COMMON.F' *****
C WRITE(*,100) TITLE
C WRITE(*,101) FILEOUT
C WRITE(*,102) LIAFIL
C WRITE(*,103) STOKFIL
C WRITE(*,104) BONDFIL
C WRITE(*,105) NSAVE
C WRITE(*,106) NSIMS
C WRITE(*,107) NSTOCKS
C WRITE(*,108) NBONDS
C WRITE(*,109) IDUMMY
C WRITE(*,110) STOCKMIN
C WRITE(*,111) YIMAX
C WRITE(*,112) SHAX
C WRITE(*,113) NSTAT
C WRITE(*,114) NRETS
C WRITE(*,115) BULLET
C WRITE(*,116) TARGET
C WRITE(*,117) TURN
C WRITE(*,118) TURN
RETURN
100 FORMAT('0',A1)

```

```

10 10 WRITE(*,*) ' '
11 11
12 12 PRINT MENU
13 13 WRITE(*,*) 'HEADER'
14 14 WRITE(*,*) ' '
15 15
16 16 DO 20 I = 1, NBROPT
17 17   WRITE(*,*) '(1X,12.2H) ',$)' I
18 18   CONTINUE
19 19
20 20 WRITE(*,*) 'LAST'
21 21
22 22 BLANK OUT REST OF SCREEN
23 23 DO 25 J = 1, IMAX
24 24   WRITE(*,*) ' '
25 25
26 26 REQUEST RESPONSE
27 27 WRITE(*,*) ' '
28 28 WRITE(*,*) ' '
29 29
30 30 WRITE(*,*) '999'
31 31
32 32 FORMAT(1X) SELECT NUMBER OF OPTION ----> ', $)
33 33 RESPONSE SET TO ZERO IN CASE OF CARRIAGE RETURN
34 34 RESPIN = 0
35 35 READ (*, '(12)', END = 40, ERR = 5) RESPON
36 36 IF (RESPON.LT.0.OR.RESPON.GT.NBROPT) THEN
37 37   WRITE(*,*) 'OPTION OUT OF RANGE'
38 38   GO TO 30
39 39
40 40 END IF
41 41 RETURN
42 42 END
43 43
44 44 C***** END OF SUBROUTINE MENU *****
45 45
46 46 C ***** SET UP PROBLEM FOR OPTIMIZER *****
47 47 SUBROUTINE SOLVE
48 48 C ***** INCLUDE 'COMMON.F' *****
49 49
50 50 WRITE(*,*) '(', 'ASO') IN SOLVE ' '
51 51 C ***** SET THE MAXIMUM SIZE FOR THE OPTIMIZER *****
52 52
53 53 D1LSTAR = 0.005
54 54 D1LSTAR = DELSTAR/1000000.
55 55 DISTMAX = 0.40
56 56
57 57 C ***** SOLVE COUNTS THE NUMBER OF RUNS, ICOUNT THE ITERATIONS *****
58 58 ISOLVE = ISOLVE + 1
59 59 ICOUNT = 0
60 60 ITEST = 0
61 61
62 62 C ***** CALL BOUNDS TO SET INDUSTRY AND SECURITY BOUNDS *****
63 63
64 64
65 65
66 66
67 67
68 68
69 69
70 70
71 71
72 72
73 73
74 74
75 75
76 76
77 77
78 78
79 79
80 80
81 81
82 82
83 83
84 84
85 85
86 86
87 87
88 88
89 89
90 90
91 91
92 92
93 93
94 94
95 95
96 96
97 97
98 98
99 99
100 100
101 101 FORMAT(A35)
102 102 FORMAT(14)
103 103 FORMAT(F7.2)
104 104 FORMAT(' ', A50)
105 105
106 106 END
107 107
108 108 ***** END OF SUBROUTINE SAVDAT *****
109 109
110 110 C***** SUBROUTINE MENU (LINES,HEADER,LAST,NBROPT,OPTION,RESPON) *****
111 111
112 112 C*****
113 113
114 114
115 115
116 116
117 117
118 118
119 119
120 120
121 121
122 122
123 123
124 124
125 125
126 126
127 127
128 128
129 129
130 130
131 131
132 132
133 133
134 134
135 135
136 136
137 137
138 138
139 139
140 140
141 141
142 142
143 143
144 144
145 145
146 146
147 147
148 148
149 149
150 150
151 151
152 152
153 153
154 154
155 155
156 156
157 157
158 158
159 159
160 160
161 161
162 162
163 163
164 164
165 165
166 166
167 167
168 168
169 169
170 170
171 171
172 172
173 173
174 174
175 175
176 176
177 177
178 178
179 179
180 180
181 181
182 182
183 183
184 184
185 185
186 186
187 187
188 188
189 189
190 190
191 191
192 192
193 193
194 194
195 195
196 196
197 197
198 198
199 199
200 200
201 201
202 202
203 203
204 204
205 205
206 206
207 207
208 208
209 209
210 210
211 211
212 212
213 213
214 214
215 215
216 216
217 217
218 218
219 219
220 220
221 221
222 222
223 223
224 224
225 225
226 226
227 227
228 228
229 229
230 230
231 231
232 232
233 233
234 234
235 235
236 236
237 237
238 238
239 239
240 240
241 241
242 242
243 243
244 244
245 245
246 246
247 247
248 248
249 249
250 250
251 251
252 252
253 253
254 254
255 255
256 256
257 257
258 258
259 259
260 260
261 261
262 262
263 263
264 264
265 265
266 266
267 267
268 268
269 269
270 270
271 271
272 272
273 273
274 274
275 275
276 276
277 277
278 278
279 279
280 280
281 281
282 282
283 283
284 284
285 285
286 286
287 287
288 288
289 289
290 290
291 291
292 292
293 293
294 294
295 295
296 296
297 297
298 298
299 299
300 300
301 301
302 302
303 303
304 304
305 305
306 306
307 307
308 308
309 309
310 310
311 311
312 312
313 313
314 314
315 315
316 316
317 317
318 318
319 319
320 320
321 321
322 322
323 323
324 324
325 325
326 326
327 327
328 328
329 329
330 330
331 331
332 332
333 333
334 334
335 335
336 336
337 337
338 338
339 339
340 340
341 341
342 342
343 343
344 344
345 345
346 346
347 347
348 348
349 349
350 350
351 351
352 352
353 353
354 354
355 355
356 356
357 357
358 358
359 359
360 360
361 361
362 362
363 363
364 364
365 365
366 366
367 367
368 368
369 369
370 370
371 371
372 372
373 373
374 374
375 375
376 376
377 377
378 378
379 379
380 380
381 381
382 382
383 383
384 384
385 385
386 386
387 387
388 388
389 389
390 390
391 391
392 392
393 393
394 394
395 395
396 396
397 397
398 398
399 399
400 400
401 401
402 402
403 403
404 404
405 405
406 406
407 407
408 408
409 409
410 410
411 411
412 412
413 413
414 414
415 415
416 416
417 417
418 418
419 419
420 420
421 421
422 422
423 423
424 424
425 425
426 426
427 427
428 428
429 429
430 430
431 431
432 432
433 433
434 434
435 435
436 436
437 437
438 438
439 439
440 440
441 441
442 442
443 443
444 444
445 445
446 446
447 447
448 448
449 449
450 450
451 451
452 452
453 453
454 454
455 455
456 456
457 457
458 458
459 459
460 460
461 461
462 462
463 463
464 464
465 465
466 466
467 467
468 468
469 469
470 470
471 471
472 472
473 473
474 474
475 475
476 476
477 477
478 478
479 479
480 480
481 481
482 482
483 483
484 484
485 485
486 486
487 487
488 488
489 489
490 490
491 491
492 492
```

128

129

```

RHS(1) = RHSMAX
DO 20 J = 1, 10
  CONS(J,1) = 0.0
C *****
  (1) = SUM OF THE WEIGHTS EQUALS 1.0 *****
  IF (J.EQ. 1) CONS(J,1) = 1.0
C *****
  (2) = SUM OF SIC CODES 48,49 LT INDUSTRY BOUND *****
  IF (J.EQ. 2) THEN
    IF (ISIC(1).EQ. 48 .OR. ISIC(1).EQ. 49) CONS(J,1) = 1.0
  END IF
C *****
  (3) = SUM OF SIC CODES 60,63,67 LT INDUSTRY BOUND *****
  IF (J.EQ. 3) THEN
    IF (ISIC(1).EQ. 60 .OR. ISIC(1).EQ. 63 .OR. ISIC(1).EQ. 67) CONS(J,1) = 1.0
  END IF
  &
C *****
  CONTINUE
20 CONTINUE
C *****
  SET RIGHT HAND SIDE FOR CONSTRAINTS AND SET TYPES *****
  RHS(NR+1) = 1.000
  RHS(NR+2) = YIMAX / 100.
  RHS(NR+3) = YIMAX / 100.
C *****
  SET UP XLHS VECTOR
  IF (IFLAG(2).EQ. 1) THEN
    DO 30 I = 1, NUM
      XLHS(I) = OLDMT(I)
    CONTINUE
  ELSEIF (IFLAG(2).EQ. 2) THEN
    DO 40 I = 1, NUM
      XLHS(I) = 0.0
    CONTINUE
  ENDIF
  RETURN
END
C *****
  END OF SUBROUTINE BOUNDS
C *****
  SUBROUTINE COVARIANCE CALCULATION SUBROUTINE *****
  SUBROUTINE COVAR *****
  *****

```

```

IF(NMSAVE.GE.8801)PORTAL=PRIVALS*
IF(NMSAVE.GE.8501)PORTAL=PRIVALS*
IF(NMSAVE.GE.8601)PORTAL=PRIVALS*
IF(NMSAVE.GE.8701)PORTAL=PRIVALS*
IF(NMSAVE.GE.8801)PORTAL=PRIVALS*
IF(NMSAVE.GE.8901)PORTAL=PRIVALS*
IF(PORTAL.EQ.0)PORTAL=100000000.0
C WRITE(*,*)'IN BOUNDS AND PORTAL = ',PORTAL
C WRITE(*,*)
C *****
  PCNT = PARAMETER FOR MAX HOLDING IN A COMPANY: 4% *****
  DAYSVOL = PARAMETER FOR MAX # TRADING DAYS IN A STOCK *****
  DO 10 I = 1, NUM
C *****
  SET UP RHS VECTOR
C *****
  RHS(1) IS THE RIGHT HAND SIDE CONSTRAINT VECTOR *****
  RHS(1) = 0.0
C *****
  SET RHSMIN EQUAL TO SMALLEST: UPPER BOUND, TRADING VOLUME,
  PERCENT OF THE COMPANY *****
  RHSMAX = X8UPPER/100.
C *****
  IF (DVOL(1).GT. 0.) THEN
    IF (DAYSVOL*1000.*DVOL(1)/PORTAL.LT. RHSMAX) THEN
      RHSMAX = DAYSVOL*1000.*DVOL(1)/PORTAL
    END IF
  END IF
  IF (XMKTCAP(1).GT. 0.) THEN
    IF ((PCNT/100.)*1000.*XMKTCAP(1)/PORTAL.LT. RHSMAX) THEN
      RHSMAX = (PCNT/100.)*1000.*XMKTCAP(1)/PORTAL
    END IF
  END IF
C *****
  SET THE MINIMUM INVESTED IN A STOCK BY USING PORTAL *****
  IF (PORTAL.GT. 1.0E+9) THEN
    RHSMIN = 0.0005
  ELSE IF (PORTAL.LT. 1.0E+8) THEN
    RHSMIN = 0.005
  ELSE
    RHSMIN = 0.0055 - (5.0E-12)*PORTAL
  END IF
  IF (RHSMAX.LT. RHSMIN) RHSMAX = 0.0
C *****

```

130

```

C ***** FIRST SORT THE COVARIANCES *****
DO 20 I = 1, NUM
  X(I) = XLHS(I)
  X0(I) = XLHS(I)
  DO 25 K = 1, ID
    IF (CONS(K,I) .EQ. 1.0) SLK(K)-SLK(K)-X(I)
  CONTINUE
DO 30 J = 1+1, NUM
  IF (COV(LI(J)) .GT. COV(LI(1))) THEN
    LITEMP = LI(1)
    LI(1) = LI(J)
    LI(J) = LITEMP
  END IF
CONTINUE
20 CONTINUE
NBEST = 0
C ***** NOW FILL THE X() VECTOR AND SWITCH THE LI() VECTOR *****
DO 40 I = 1, NUM
  ***** FILL X(I) IF NOT CONSTRAINED *****
  SLKMAX = RHS(LI(J))-XLHS(LI(J))
  ***** FIND THE TIGHTEST CONSTRAINT *****
  DO 50 K = 1, ID
    IF (CONS(K,LI(J)) .EQ. 1.0) THEN
      IF (SLK(K) .LT. SLKMAX) THEN
        SLKMAX = SLK(K)
      END IF
    END IF
  END IF
CONTINUE
50
C ***** SET THE VARIABLE = TO THE MOST SLACK AVAILABLE *****
IF (SLKMAX .LT. 0.00001) GO TO 40
C ***** NOW REDUCE THE SLACK IN EACH CONSTRAINT *****
DO 60 K = 1, ID
  IF (CONS(K,LI(J)) .EQ. 1.0) THEN
    SLK(K) = SLK(K) - SLKMAX
  END IF
CONTINUE
60
NBEST = NBEST + 1
LITEMP = LI(NBEST)
LI(NBEST) = LI(J)
LI(J) = LITEMP
X(LI(NBEST)) = SLKMAX + X(LI(NBEST))
X0(LI(NBEST)) = SLKMAX + X(LI(NBEST))
IF (SLK(1) .LE. 0.00001) RETURN

```

```

      INCLUDE 'COMMON.F'
      WRITE(*, '(+,.A50)') 'IN COVAR .....
C ***** INITIALIZE LI(1), CALC AVE(1), CALC COV(1) *****
C ***** UANIM = 0.0
      UANIM = 0.0
      VARANIM = 0.0
C ***** INITIALIZE LI(1) VECTOR *****
DO 10 I = 1, NUM
  LI(1) = 1
  AVE(1) = 0.0
  COV(1) = 0.0
DO 20 J = 1, NSTAT
  IF (1 .EQ. 1) THEN
    UANIM = UANIM + ANIM(J)*TARGET
    VARANIM = VARANIM + ANIM(J)*ANIM(J)*TARGET
  END IF
  IF (RET(J,1) .LT. -1.0) RHS(1) = 0.0
  AVE(1) = AVE(1) + RET(J,1)
  COV(1) = COV(1) + RET(J,1)*ANIM(J)*TARGET
CONTINUE
IF (1 .EQ. 1) THEN
  UANIM = UANIM / REAL(NSTAT)
  VARANIM = VARANIM / REAL(NSTAT)
  UANIM = UANIM
  AVE(1) = AVE(1) / REAL(NSTAT)
  COV(1) = COV(1) / REAL(NSTAT)
CONTINUE
10 CONTINUE
RETURN
END
C ***** CALCULATE STARTING POINT *****
C ***** SUBROUTINE STARTPT *****
C ***** THIS SUBROUTINE CALCULATES A STARTING POINT.
C ***** THE STARTING POINT IS THEN USED FOR PARTIAL.
      INCLUDE 'COMMON.F'
      WRITE(*, '(+,.A50)') 'IN STARTPT ....'
      DO 10 I = 1, ID
        SLK(I) = RHS(NUM+1)
      CONTINUE

```





132

```

50      CONTINUE
60      CONTINUE
C ***** NOW SORT THE PARTIALS WITH ZERO WEIGHTS *****
NBEST = 0
DO 60 I = 1, NUMB
    IBEST = 0
    DO 70 K = 1, ID
        IF (CONS(K,LI(1))) .EQ. 1.0 .AND. SLK(K) .LT. 100.*EPS) THEN
            IF (PART(LI(1))) .GT. RC(K) IBEST = I
        END IF
    END IF
    CONTINUE
    IF (IBEST .EQ. 0) THEN
        NBEST = NBEST + 1
        LITEMP = LI(NBEST)
        LI(NBEST) = LI(1)
        LI(1) = LITEMP
    END IF
70      CONTINUE

C ***** IF THIS IS A FULL PARTIAL --> DO A FULL SORT *****
C ***** IF (LITEST .EQ. 0) THEN DO A FULL SORT OF NUMERICAL RECIPES *****
WRITE(*,('A12')) IN SORT_P ...
LHEAP = NBEST/2 + 1
IR = NBEST
CONTINUE
IF (LHEAP.GT.1) THEN
    LHEAP = LHEAP - 1
    INDXT = LI(LHEAP)
    Q = PART(INDXT)
    ELSE
        INDXT = LI(IR)
        Q = PART(INDXT)
        LI(IR) = LI(1)
        IR = IR - 1
        IF (IR .EQ. 1) THEN
            LI(1) = INDXT
            GOTO 111
        ENDIF
    ENDIF
    LHEAP = LHEAP + LHEAP
    IF (JHEAP.LE.IR) THEN
        IF (JHEAP.LI(1)) THEN
            IF (PART(LI(JHEAP))) .LT. PART(LI(LHEAP+1)) JHEAP = JHEAP + 1
        ENDIF
        IF (Q.LT.PART(LI(JHEAP))) THEN

```

```

DO 20 I = 1, NUMB
    IF (X(LI(1))) .GT. 0.0) THEN
        DO 30 K = 1, ID
            IF (CONS(K,LI(1))) .EQ. 1.0) SLK(K) = SLK(K) - X(LI(1))
        CONTINUE
    END IF
30      CONTINUE

    PART(LI(1)) = 0.0
    RAVE = 0.0
    IF (IFLAG(4) .EQ. 2) THEN
        DO 35 J = 1, NSTAT
            XRET = (1+RET(J,LI(1))) * (1-RET(J-1,LI(1))) * (1+RET(J-2,LI(1))) - 1
            XRET = XRET - (1+ANIM(J-1)) * (1+ANIM(J-1)) * (1+ANIM(J-2)) - 1 * TARGET
            PART(LI(1)) = PART(LI(1)) + XRET * PRET(J)
            RAVE = RAVE + XRET / REAL(NSTAT-2)
        CONTINUE
35      CONTINUE

    PART(LI(1)) = 2 * (PART(LI(1)) / REAL(NSTAT-2) - RAVE * PAVE) - RAVE * BULLET
    END IF
C *****
C ***** END MOVING QUARTERLY
C ***** IF MONTHLY RETURNS
    IF (IFLAG(4) .EQ. 1) THEN
        DO 40 J = 1, NSTAT
            PART(LI(1)) = PART(LI(1)) + (RET(J,LI(1)) - ANIM(J) * TARGET) * PRET(J)
            RAVE = RAVE + (RET(J,LI(1)) - ANIM(J) * TARGET) / REAL(NSTAT)
        CONTINUE
40      CONTINUE

    PART(LI(1)) = 2.0 * (PART(LI(1)) / REAL(NSTAT) - RAVE * PAVE) - RAVE * BULLET
    END IF
C *****
C ***** IF TURNOVER IS A FACTOR ADD IN A COST
    IF (TURN .GT. 0.0) THEN
        PARTURN = TURN * (X(LI(1)) - OLDWT(LI(1))) /
        ((X(LI(1)) - OLDWT(LI(1))) ** 2 + 1.0E-7) ** 0.5
        IF (OLDWT(LI(1)) .EQ. 0.0) PARTURN = TURN
        PART(LI(1)) = PART(LI(1)) + PARTURN
    ENDIF
C *****
C ***** FIND WORST PARTIAL FOR A POSITIVE WEIGHT STOCK
    DO 50 K = 1, ID
        IF (X(LI(1))) .GT. EPS) THEN
            IF (CONS(K,LI(1))) .EQ. 1.0 .AND. PART(LI(1)) .GT. RC(K)
                RC(K) = PART(LI(1))
        END IF
50      CONTINUE

```

```

20      IF (SLK(K).LT.SLKMAX ) SLKMAX = SLK(K)
      END IF
      CONTINUE
C *****
      FIND LARGEST MOVE POSSIBLE *****
      DIST = DISTXI
      IF (DIST .LT. 0) WRITE(*,*)'1'
      DP = (PART(LI(IB)) - PART(LI(1))) * 10.
      IF (DP .LT. DIST) DIST = DP
      IF (SLKMAX .LT. DIST) DIST = SLKMAX
      IF (X(LI(IB)) - XLHS(LI(IB)).LT.DIST) DIST = X(LI(IB)) - XLHS(LI(IB))
      IF (RHS(LI(1)) - X(LI(1)).LT.DIST) DIST = RHS(LI(1)) - X(LI(1))
      IF (TURN .GT. 0.0) THEN
      IF (XO(LI(1)) .LT. OLDWT(LI(1))) .AND.
      & OLDWT(LI(1)) - X(LI(1)) .LT. DIST )DIST = OLDWT(LI(1)) - X(LI(1))
      IF (XO(LI(IB)) .GT. OLDWT(LI(IB))) .AND.
      & X(LI(IB)) - OLDWT(LI(IB)) .LT. DIST )DIST = X(LI(IB)) - OLDWT(LI(IB))
      END IF
      IF (DIST .LT. 0.0001*EPS) GO TO 5
      IF (DP .LT. 0.1*PHAX) GO TO 5
      IF (DP .GT. PHAX) PHAX = DP
C *****
      CHANGE THE X VECTOR AND THE CONSTRAINTS *****
      X(LI(1)) = X(LI(1)) - DIST
      X(LI(1)) = X(LI(1)) + DIST
      DO 30 K = 1, ID
      IF (CONS(K,LI(1))) .EQ. 1.0 ) SLK(K) = SLK(K) + DIST
      IF (CONS(K,LI(1))) .EQ. 1.0 ) SLK(K) = SLK(K) - DIST
      IF (SLK(K) .LT. EPS) SLK(K) = 0.0
      CONTINUE
30
C *****
      TEST FOR TOTAL MOVE EXCEEDING DISTMAX *****
      MOVES = MOVES + 1
      TOTDIST = TOTDIST + DIST
      IF (TOTDIST .LT. 0.0 .OR. MOVES .GE. 15) GO TO 99
C *****
      IF THE WORST VARIABLE STILL HAS SLACK, REDUCE IT *****
      IF (ABS(XO(LI(1)) - X(LI(1))) .LT. 0.5*DISTXI ) GO TO 5
10      CONTINUE
      IF (SLK(K).LT.SLKMAX ) SLKMAX = SLK(K)
      IHEAP = IHEAP - JHEAP
      JHEAP = JHEAP + JHEAP
      ELSE JHEAP = IR + 1
      ENDIF
      GOTO 120
      ENDIF
      LI(IHEAP) = INDXT
      GOTO 110
111      CONTINUE
      END IF
C *****
      OBJ2 = OBJ
      RETURN *****
C *****
      END OF PARTIAL *****
      MOVE TO A NEW POINT BASED ON PARTIALS *****
      SUBROUTINE MOVE_X *****
      ***** THIS SUBROUTINE MOVES FROM CURRENT POINT TO NEW POINT *****
      *****
      INCLUDE 'COMMON.F'
      PARAMETER ( EPS = 0.000001)
      WRITE(*,*)('+',A12) 'IN MOVE_X ...'
      DISTMAX = DISTMAX + 1.05
      IF (DISTMAX .GT. 0.4) DISTMAX = 0.4
      DISTXI = 0.15 * DISTMAX
      MOVES = 0
      TOTDIST = DISTMAX
      PHAX = -9999.
      *****
      DO 10 I = 1, NBEST
      IF (RHS(LI(1)) - X(LI(1))) .LT. EPS) GO TO 10
      IB = NBEST + 1
      IF (IB = 1) GO TO 10
      IF (PART(LI(IB)) .LE. PART(LI(1))) GO TO 10
      *****
      TEST PROXIMITY TO RHS AND TOTAL MOVEMENT *****
      IF (X(LI(1)) - XLHS(LI(1))) .LT. EPS) GO TO 5
      IF (ABS(XO(LI(1)) - X(LI(1))) .GT. 0.5*DISTXI) GO TO 5
      *****
      TEST FOR CONSTRAINT PROXIMITY *****
      SLKMAX = 1.0
      DO 20 K = 1, ID
      IF (CONS(K,LI(1)) .EQ. 1.0 .AND. CONS(K,LI(1)) .EQ. 0.0) THEN
      IF (SLK(K).LT.EPS) GO TO 5

```

C \*\*\*\*\*IF THIS IS A SINGLE RUN READ WEIGHT FILE SPECIFIED

[illegible]

135

```

21      ENDIF
      CONTINUE

      IF(OLDTOT.LT..99)THEN
        WRITE(*,*)' WARNING OLDWTS TOTAL < .99, CONTINUING '
        WRITE(*,*)' OLDTOT = ',OLDTOT
      ELSEIF(OLDTOT.GT.1.01)THEN
        WRITE(*,*)' WARNING OLDWTS TOTAL > 1.01, CONTINUING '
        WRITE(*,*)' OLDTOT = ',OLDTOT
      ENDIF

      NGONE=NCOUNT
      IF (IDCTEST.EQ.1) THEN
        WRITE(*,*)' A TOTAL OF ',VALTURN*100.0, ' % OF PORT SOLD'
        WRITE(*,*)NGONE,'SYMLS NOT FOUND, PROGRAM CONTINUING'
      ENDIF

C      IF THE 1ST TIME THROUGHT SET OLDWT()=0

      ELSEIF(ISKIP.EQ.1)THEN
        DO 98 I=1,NUM
          OLDWT(I)=0.0
98      CONTINUE
      ENDIF

C      ***** IF MULTI RUN (&OLD) READ WTS FRO OLD.

      ELSEIF(IFLAG(1).EQ.3)THEN
C      OPEN(21,FILE = 'OLD',STATUS = 'OLD')
C      DO 94 I = 1,NUM
C        READ(21,*)OLDWT(I)
C        IF (TURN.EQ. 0.0)OLDWT(I)=0.0
C        THIS NEXT LINE ADDED 4/18/89
C        OLDWT(I)=0.0
C        OLDWT(I) = OLDWT(I)/100.
94      CONTINUE
C      CLOSE(21)

      ENDIF

      RETURN

      END

C *****
      END OF SUBROUTINE HOLDING *****

```

137

APPENDIX V

REINVESTMENT OF AVAILABLE CASH  
BEGINNING FROM CURRENT  
PORTFOLIO HOLDINGS

138

IMPROVED SYTEN ANALYSIS: RE-INVEST DIVIDEND FLOWS  
SIMULATION RESULTS

EXAMPLE RUN

LIABILITY STREAM USED: EXAMPLE LIABILITY

295 2.372 2.372 1.035 73 SHED SHED MED SYS CORP COM  
 296 1.889 1.889 1.086 38 MES STR QUESTAR CORP COM  
 297 2.299 2.299 1.353 1 TEJN TRC TEJON RANCH CO DEL COM  
 298 3.628 3.628 1.138 80 USHC USHC U S HEALTHCARE INC COM  
 299 3.406 3.406 1.061 38 USSC USS UNITED STATES SURGICAL CO  
 300 1.375 1.375 0.947 49 MGL MGL WASHINGTON GAS LT CO COM  
 301 1.844 1.844 0.995 67 MRE MRE WASHINGTON REAL ESTATE IN  
 302 1.262 1.262 0.930 49 MFR5 MCN MCN CORP COM

MINIMUM STD DEV - 18.8463 EXPECTED RETURN - 17.6218  
 TRACKER OUTPUT FILE

MONTH	LIA	PORT	DIF
115	-0.005	0.049	

## CUMULATIVE VALUES OF LIABILITIES AND ASSETS

MONTH LIABILITIES PORTFOLIO  
 115 -0.005 0.049  
 \*\*\*\*\* STATISTICAL SUMMARY \*\*\*\*\*

STATISTICS BASED ON MONTHS 116 THROUGH 115

	ALPHA	BETA	STD ERR	R-SQUARED
ANNUAL RETURNS:	0.000000	0.000000	0.000000	0.000000
LIABILITY	0.000000	0.000000	0.000000	0.000000
PORTFOLIO	0.000000	0.000000	0.000000	0.000000
DIFFERENCE	0.000000	0.000000	0.000000	0.000000

	MEAN	STD DEV
ANNUAL RETURNS:	0.00 %	0.00 %
LIABILITY	0.00 %	0.00 %
PORTFOLIO	0.00 %	0.00 %
DIFFERENCE	0.00 %	0.00 %

## PARAMETERS FOR THIS RUN WERE:

1) SPANNING RUN TITLE AOS 7/28/89 RUN  
 2) OUTPUT FILENAME temp.nib  
 3) LIABILITY STREAM FILE \LIR\AOS725.LIR  
 4) STOCK RETURN FILE \RET\AOSUP.RET  
 5) BOND RETURN FILE \RET\BONDS.PRN  
 6) WEIGHT FILE \WGT\159.WGT  
 7) MONTH SIMULATION BEGINS 8908  
 8) NUMBER OF MONTHS SIMULATED 302  
 9) NUMBER OF STOCKS 0  
 10) MINIMUM STOCK HOLDING % 100.00  
 11) MINIMUM INDUSTRY HOLDING % 19.00  
 12) MAXIMUM INDUSTRY HOLDING % 19.00

NUMBER	WGT	SPAN	SIC	IDC	TICK	SECURITY NAME
45	3.475	3.475	1.024	27	GEN	GTE CORP COM
62	1.693	1.693	1.088	48	SBCA	SOUTHWESTERN BELL CORP COM
74	2.196	2.196	1.194	37	BA	BOEING CO COM
82	3.508	3.508	1.021	20	K	KELLOGG CO COM
136	0.031	0.031	1.052	56	LINT	LTD LIMITED INC COM
194	0.041	0.041	1.638	34	MAS	MASCO CORP COM
231	1.475	1.475	0.702	67	FBG1	BANC ONE CORP COM
243	2.680	2.680	1.295	29	PZL	PENNZOIL CO COM
252	2.410	2.410	0.883	49	DTE	DETROIT EDISON CO COM
260	1.204	1.204	0.926	49	BGE	BALTIMORE GAS & ELEC CO COM
261	2.835	2.835	1.069	81	FUN	FIRST UN CORP COM NASD BK INS
262	0.929	0.929	1.276	29	ASH	ASHLAND OIL INC COM
263	1.546	1.546	1.030	28	W	WESTVACO CORP COM
264	1.399	1.399	0.973	49	MPC	WISCONSIN ENERGY CORP COM
265	1.558	1.558	0.773	60	AHM	ARMANSON H F & CO COM
266	2.761	2.761	1.043	12	PEL	PANHANDLE EASTN CORP COM
267	2.690	2.690	0.887	49	IPC	ILLINOIS POWER CO COM
268	2.726	2.726	0.931	67	NES	NEW ENGLAND ELEC SYS COM
269	0.563	0.563	0.957	49	TSE	TUCSON ELEC PWR CO COM
270	1.142	1.142	1.068	7	IFLH	IFM FERTILIZER GROUP INC COM
271	1.187	1.187	1.070	75	AGNC	AGENCY RENT A CAR COM NASD INDL
272	0.819	0.819	0.897	49	ATA	ATLANTA GAS LT CO COM
273	2.698	2.698	1.082	30	BDG	BANDAG INC COM
274	0.451	0.451	1.349	32	BLL	BALL CORP COM
275	2.909	2.909	0.882	49	BSE	BOSTON EDISON CO COM
276	1.155	1.155	0.909	67	CALH	CALFED INC COM
277	1.700	1.700	1.058	73	CMCS	COMCAST CORP CL A NASD INDL
278	2.093	2.093	1.678	35	LANC	3COM CORP COM NASD INDL
279	1.190	1.190	1.131	67	FPA	FIRST PA CORP COM
280	3.346	3.346	0.754	67	HCP	HEALTH CARE PPTY INVS INC COM REIT
281	2.418	2.418	1.036	8	IPT	IP TIMBERLAND LTD DEP UNIT CL A
282	2.442	2.442	1.044	27	HJH	HARLAND JOHN H CO COM
283	0.182	0.182	0.935	49	KAN	KANSAS PWR & LT CO COM
284	2.694	2.694	0.896	65	KOG	KOGER PPTYS INC COM REI
285	0.535	0.535	1.152	20	LNE	LANCE INC COM NASD INDL
286	3.172	3.172	0.885	52	LCL	LONES COS INC COM
287	3.207	3.207	0.895	45	SARI	SOUTHWEST AIRLS CO COM
288	3.000	3.000	0.967	67	MTRV	MEDITRUST SH BEN INT
289	1.238	1.238	0.904	49	IPS	MIDWEST ENERGY CO COM
290	3.005	3.005	1.184	36	NPX	NATIONAL PRESTO INDS INC COM
291	2.755	2.755	0.994	67	RCPA	ROCKEFELLER CITR PPTYS INC COM REIT
292	0.846	0.846	1.181	23	RHL	RUSSELL CORP COM
293	1.158	1.158	1.157	38	SEFC	SEED GENETICS CORP CL B NASD INDL
294	1.256	1.256	0.889	49	STG	SOUTHERN IND GAS & ELEC CO COM



13) MAXIMUM STOCK HOLDING % 3.00 %  
 14) INSTAT 24  
 15) NRETS 24  
 16) BULLET 0.00  
 17) TARGET 2.00  
 18) TURNOVER FACTOR 4.00  
 19) MAX DAYS VOLUME 26.00  
 20) PORTFOLIO VAL (89 MM)  
 21) MAX % STK MKT CAP 3.00  
 22) MIN MKT CAP (89 MM) 2500.00  
 23) MAX MKT CAP (89 MM) 1000000.00  
 24) SINGLE  
 25) 1 WAY BUY 1  
 26) SPAN 1  
 27) MONTHLY 1  
 28) FULL COVARIANCE 1  
 29) FORMATTED FILES 1  
 30) STATUS IS OK 1  
 31) \*WTS OUTPUT ONLY 2  
 32) STOCKS TO SCREEN 1

## PORTFOLIO CONSTRUCTED UPON THE FOLLOWING DATA:

MONTH	LIA	PORT	DIF
92-0	0.235312600	0.0048044603	0.0283357203
93-0	0.0558954108	-0.0405714661	0.0151239447
94	0.0943555534	-0.1450983137	0.2394538671
95	0.0001456921	-0.0072849635	-0.0074306554
96	0.0176604499	0.0324543230	0.0145938732
97	0.0754733980	0.0913652703	0.0158918723
98	0.0176621992	0.0246764552	0.0070142560
99-0	0.0450356193	-0.0050582327	0.0399773866
100-0	0.0287487004	-0.0243650582	0.0043836422
101-0	0.0128774298	0.0054894658	0.0183668956
102	0.0561982282	0.058000170	-0.0003982112
103-0	0.0344604515	-0.0200813692	0.0143790822
104-0	0.0011062390	0.0000485850	0.0011548240
105	0.0447797514	0.0430007204	-0.0017790310
106	0.0371330716	0.0263595097	-0.0107735619
107-0	0.0318946689	-0.0176053215	0.0142893475
108	0.0158855300	0.0229464527	0.0070609227
109	0.0271685901	0.0407870896	0.0136204995
110-0	0.0267152898	0.0007467365	0.0274620261
111	0.0109503297	0.0284809042	0.0175305754
112	0.0262673292	0.0394087993	0.0131414700
113	0.0461410606	0.0458422191	-0.0002988614
114	0.0783677101	0.0054350942	-0.0729326159
115-0	0.0048000002	0.0486113280	0.0534113273

TURNOVER = 0.91 %  
 TOTAL PURCHASE = 1.83 %

TOTAL OPTIONAL SALES = 0.00 %  
 TOTAL FORCED SELLS = 0.00 %

# OF STOCKS IN PORTFOLIO = 52  
 NUMBER OF BUYS = 14  
 NUMBER OF SELLS = 0

VARIANCE AT OLD WEIGHTS = 18.63 %  
 VARIANCE AT NEW WEIGHTS = 18.85 %  
 OBJ FUNCTION IMPROVEMENT = -1.15 %

THIS YRS PORTFOLIO VALU = \$ 28062494.00  
 1989 EQUIVALENT VALUE = \$ 28062494.00

*** BUY ***			
OLD NEW	WEIGHTS **	SPAN SENS	CODE
2.88	2.91	0.03	8.821
2.39	2.41	0.02	8.828
1.57	2.69	1.12	8.873
0.00	0.04	0.0410	3.77
2.74	2.76	0.0210	4.31
0.00	0.03	0.0310	5.21
1.66	1.70	0.0410	5.79
1.13	1.14	0.0210	6.76
1.39	1.89	0.5010	8.60
TOTAL BUY \$ 512213.			

*** SELL ***			
OLD NEW	WEIGHTS **	SPAN SENS	CODE
4.	17.000	0.0	BOSTON EDISON CO
2.	21.125	0.0	DETROIT EDISON CO
189.	16.625	00.0	ILLINOIS POWER CO
4.	30.125	0.0	MASCO CORP COM
2.	27.750	0.0	PANHANDLE EASTN CO
3.	34.875	0.0	LIMITED INC COM
4.	25.625	0.0	COMCAST CORP CL A
1.	35.125	0.0	IMC FERTILIZER GRO
37.	37.500	0.3	QUESTAR CORP COM
TOTAL SELL \$ 512213.			

*** INDUSTRY WEIGHTINGS ***			
CODE	INDUSTRY	WEIGHT	
1	AGRICULTURAL PRODUCTION-CROPS	2.30 %	
7	AGRICULTURAL SERVICES	1.14 %	
8	FORESTRY	2.42 %	
12	BITUMINOUS COAL & LIGNITE MIN.	2.76 %	
20	FOOD KINDRED PRODUCTS	4.04 %	
23	APP. & OTH. FIN. PROD. MFFAOSH	0.85 %	
26	PAPER AND ALLIED PRODUCTS	1.55 %	
27	PRINTING PUBLISHING AND A.P.	5.92 %	

29	PETROLEUM REFIN. & REL. PROD.	3.21 %
30	RUBBER AND MISC. PLAST. PROD.	2.70 %
32	STONE CLAY GLASS & CONC. PROD.	0.45 %
35	MACHINERY EXCEPT ELECTRICAL	2.09 %
36	ELE. AND ELE. MACH.	3.00 %
37	TRANSPORTATIONS EQUIPMENT	2.20 %
38	MEAS. ANAL. & CONT. INST. ETC.	6.45 %
45	TRANSPORTATION BY AIR	3.21 %
48	COMMUNICATION	1.69 %
49	ELECTRIC GAS AND SANIT. SERV.	17.31 %
52	BUILDING MAT. HARD. GAR. SUPP.	3.17 %
60	BANKING	1.56 %
61	CRED. AGEN. OTH. THAN BANKS.	2.83 %
65	REAL ESTATE	2.69 %
67	HOLD. AND OTHER INV. COMP.	17.49 %
73	BUSINESS SERVICES	4.07 %
75	AUTO. REPAIR SERV. AND GAR.	1.19 %
80	HEALTH SERVICES	3.63 %
	TOTAL -	99.99 %

APPENDIX VI

EXAMPLE ILLUSTRATING DISTINCTIONS BETWEEN

CURRENT SYSTEM AND PRIOR ART

Prior methods have required the calculation of a covariance matrix. Here the covariance matrix is calculated to advantage of the current system. The covariance cells evaluated standard definition and the cells in the matrix are filled. The calculations produce the following array:

Covariance Matrix

	x1	x2	x3
x1	0.0525	0.0275	-0.0162
x2	0.0275	0.0525	0.0188
x3	-0.0162	0.0188	0.0525

The prior methodology calculates the partial derivative by weights times the entries in each row and multiplying the total by partial derivative can be calculated for x2 at the current point following manner:

$$\text{Partial}(x2) = 2 * (0.3 * 0.0275 + 0.5 * 0.0525 + 0.2 * 0.0) = 0.0765$$

The current system calculates the partial derivative by finding portfolio returns at the current weights for the 8 periods and the covariance of x2 with the portfolio. The return in period(1) portfolio is:

$$\text{Portfolio return}(1) = 0.3 * 0.8 + 0.5 * 0.7 + 0.2 * 0.4 = 0.67$$

The other returns for periods 2 through 7 are calculated for portfolio. Then covariance between x2 and the portfolio is simply in the standard way. The result is the partial derivative of x2:

$$\text{Covariance (portfolio, x2)} = 0.0765$$

To illustrate the current system's approach to solving an appropriate problem, a simple example is outlined to point out distinctions and improvements over prior art. The example problem objective function is to find the minimum variance portfolio relative to a standard financial target. The constraints are that the sum of the security weights must be 1.0 and the lower bound of 0.0 on each security. This example will illustrate the extraction of a search direction and outline the optimization process.

Beginning with a series of eight returns for three securities, x1, x2, and x3, the average returns are calculated for each security. The eight returns represent monthly total returns less the return on a standard financial target for each of the securities in this simple example.

Net Security Returns After Subtracting Target Returns

Period	x1	x2	x3
1	0.8	0.7	0.4
2	0.7	0.8	0.7
3	0.6	0.1	0.2
4	0.4	0.6	0.6
5	0.1	0.3	0.8
6	0.2	0.4	0.5
7	0.5	0.5	0.1
8	0.3	0.2	0.3
Average	x1	x2	x3
	0.45	0.45	0.45

This result is identically the same as extracting the partial derivative from the full covariance matrix. However, fewer calculations are required and less computer memory is needed to obtain the partial derivatives. Mathematically, the results prove to be exactly the same.

Once the partial derivatives have been extracted for each of the variables, it is a simple matter to search for a better solution using the partials to indicate a direction which improves the objective function. The simple process of letting the partial derivatives guide a search, recalculating the partials at the new solution, and searching again, leads quickly to an optimal solution. The search proceeds downhill until the objective function can no longer improve. This is the optimal solution.

In summary, the current system requires fewer calculations to determine an optimal point and therefore can solve problems in much less time. The current system also requires much less computer memory and therefore can solve much larger problems. By analyzing more securities at once, a better solution can be achieved.

143

OUTPUT FROM PROGRAM

144

NATIONAL INVESTMENT SERVICES SPANNING TECHNOLOGY  
SIMULATION RESULTS

EXAMPLE UPDATE WITH CONTROLLED TURNOVER

LIABILITY STREAM USED

NUMBER	WGT	END	SPAN	SENSIT	SIC	IDC	TICK	SECURITY	NAME
1	3.365	3.365	0.000	0.000	35	LANS	COMS	3COM	CORP NASD INDI
2	0.992	0.992	0.001	0.001	75	AGNC	AGNC	AGENCY RENT A CAR	NASD INDI
3	1.414	1.414	0.000	0.000	60	AMN	AMN	AMERSON H F & CO	NASD INDI
4	1.353	1.353	0.001	0.001	48	T	ASH	AMERICAN TEL & TELEG CO	NASD INDI
5	0.683	0.583	0.002	0.002	29	ASH	ASH	ASHLAND OIL INC	NASD INDI
6	1.007	1.007	0.001	0.001	49	ATA	ATA	ATLANTA GAS LT CO	NASD INDI
7	1.217	1.217	0.001	0.001	32	BALL	BLI	BALL CORP	NASD INDI
8	1.418	1.418	0.001	0.001	49	BGE	BGE	BALTIMORE GAS & ELEC CO	NASD INDI
9	1.199	1.199	0.001	0.001	60	FBI	ONE	BANC ONE CORP	NASD INDI
10	3.023	3.023	0.002	0.002	30	BDE	BDE	BANDAG INC	NASD INDI
11	2.154	2.154	0.001	0.001	37	BA	BA	BOEING CO	NASD INDI
12	3.120	3.120	0.001	0.001	49	BSE	BSE	BOSTON EDISON CO	NASD INDI
13	1.397	1.397	0.001	0.001	35	BGG	BGG	BRIGGS & STRATTON CORP	NASD INDI
14	1.276	1.276	0.000	0.000	67	CALH	CAL	CASTLE & COOKE INC	NASD INDI
15	0.344	0.344	0.001	0.001	20	CKE	CKE	COMCAST CORP	NASD INDI
16	0.616	0.616	0.001	0.001	73	CHCS	CHCS	COMMODORE INTL LTD	NASD INDI
17	0.414	0.414	0.001	0.001	35	CBUI	CBU	COSTCO WHSL CORP NASD INDI	NASD INDI
18	0.181	0.181	0.002	0.002	51	DEK1	DKLB	DEKALB CORP NASD INDI	NASD INDI
19	1.486	1.486	0.003	0.003	51	FPA	FPA	FIRST PA CORP	NASD INDI
20	2.015	2.015	0.002	0.002	60	FUN	FUN	FIRST UN CORP	NASD INDI
21	2.628	2.628	0.001	0.001	61	GEN	GTE	GTE CORP	NASD INDI
22	3.294	3.294	0.001	0.001	67	HCP	HCP	HEALTH CARE PPTY INVS INC	NASD INDI
23	3.000	3.000	0.001	0.001	8	IPT	IPT	IP TIMBERLAND LTD	NASD INDI
24	2.811	2.811	0.002	0.002	49	KAN	KAN	KANSAS PWR & LT CO	NASD INDI
25	0.211	0.211	0.001	0.001	20	K	K	KELLOGG CO	NASD INDI
26	3.002	3.002	0.001	0.001	65	KOG	KOG	KOGER PPTYS INC	NASD INDI
27	3.000	3.000	0.002	0.002	20	LNC	LNC	LANCE INC	NASD INDI
28	0.251	0.251	0.002	0.002	52	LCL	LOW	LOWES COS INC	NASD INDI
29	1.827	1.827	0.001	0.001	67	MTR	MT	MEDITRUST SH BEN INT	NASD INDI
30	2.944	2.944	0.002	0.002	49	MFR5	MCN	MICHIGAN CONS GAS CO	NASD INDI
31	1.280	1.280	0.001	0.001	49	IPS	MME	MIDWEST ENERGY CO	NASD INDI
32	1.438	1.438	0.001	0.001	35	NPK	NPK	NATIONAL PRESTO INC	NASD INDI
33	3.045	3.045	0.001	0.001	67	NES	NES	NEW ENGLAND ELEC SYS	NASD INDI
34	3.100	3.100	0.000	0.000	25	OSH	OMT	OHIO MATTRESS CO DEL	NASD INDI
35	1.098	1.098	0.001	0.001	48	PACB	PAC	PACIFIC TELESIS GROUP	NASD INDI
36	1.445	1.445	0.001	0.001	12	PEL	PEL	PANHANDLE EASTIN CORP	NASD INDI
37	2.879	2.879	0.003	0.003	29	PZL	PZL	PENNZOIL CO	NASD INDI
38	2.955	2.955	0.002	0.002	13	PMK	PMK	PRIMARK CORP	NASD INDI
39	0.496	0.496	0.001	0.001	38	MES	STR	QUESTAR CORP	NASD INDI
40	3.010	3.010	0.001	0.001	67	RCPA	RCP	ROCKEFELLER CTR PPTYS INC	NASD INDI
41	3.025	3.025	0.001	0.001	37	RHR	RHR	ROHR INDS INC	NASD INDI
42	0.175	0.175	0.002	0.002	37	RHR	RHR	ROHR INDS INC	NASD INDI

43	0.170	0.570	0.001	23	RML	RML	RUSSELL CORP	NASD INDI
44	3.155	3.155	0.000	73	SMED	SMED	SHARED MED SYS CORP	NASD INDI
45	1.177	1.377	0.001	49	SIG	SIG	SOUTHERN IND GAS & ELEC CO	NASD INDI
46	2.632	2.632	0.000	45	SARI	LUV	SOUTHWEST AIRLCS CO	NASD INDI
47	1.569	1.569	0.001	48	SBCA	SBC	SOUTHWESTERN BELL CORP	NASD INDI
48	2.201	2.201	0.001	1	TEJN	TRC	TUCSON RANCH CO	NASD INDI
49	1.291	1.291	0.001	49	TSB	TEP	TUCSON ELEC PWR CO	NASD INDI
50	2.246	2.246	0.001	80	USHC	USHC	U S HEALTHCARE INC	NASD INDI
51	2.118	2.118	0.000	49	UIL	UIL	UNITED ILLUM CO	NASD INDI
52	3.173	3.173	0.002	38	USSC	USSC	UNITED STATES SURGICAL	NASD INDI
53	1.488	1.488	0.001	49	WGL	WGL	WASHINGTON GAS LT CO	NASD INDI
54	1.582	1.582	0.001	49	WPC	WEC	WISCONSIN ENERGY CORP	NASD INDI
55	0.583	0.583	0.002	51	SEED	SEED	DEKALB GENETICS CORP	NASD INDI
56	0.229	0.229	0.002	51	PTVZ	DKLB	PRIDE PETE SVCS INC	NASD INDI

MINIMUM STD DEV = 73.0185 EXPECTED RETURN = 9.0049

TRACKER OUTPUT FILE

MONTH	LIA	PORT	DIF
-------	-----	------	-----

CUMULATIVE VALUES OF LIABILITIES AND ASSETS

MONTH	LIABILITIES	PORTFOLIO
-------	-------------	-----------

107 0.022 -0.005

\*\*\*\*\* STATISTICAL SUMMARY \*\*\*\*\*

STATISTICS BASED ON MONTHS 108 THROUGH 107

ALPHA	0.000000	ANNUAL	0.00 %
BETA	0.000000		
STD ERR	0.000000		
CORRELATION	0.000000		
R-SQUARED	0.000000		

ANNUAL RETURNS:

LIABILITY	0.00 %	STD DEV	0.00 %
PORTFOLIO	0.00 %		
DIFFERENCE	0.00 %		

PARAMETERS FOR THIS RUN WERE:

- 1) SPANNING RUN TITLE UPDATE A.O. STEEL 12 5 88
- 2) OUTPUT FILENAME TEMP.SPN
- 3) LIABILITY STREAM FILE \LIR\AOSTEEL-LIR
- 4) STOCK RETURN FILE \RET\AOSNOV.RET

145

\*\*\*\*\*  
 \*\*\* WEIGHTS \*\*\* SPAN SIC SYMBOL SECURITY  
 OLD NEW DELTA SENSIT CODE IDC TICK NAME  
 \*\*\*\*\*  
 2.847 3.000 0.153 2.124 65 KOG KOG KOG PPTYS INC  
 2.680 2.944 0.265 2.508 67 MTRV MT MEDITRUST SH BEN INT  
 \*\*\*\*\*  
 \*\*\* SELL \*\*\*  
 \*\*\*\*\*  
 \*\*\* WEIGHTS \*\*\* SPAN SIC SYMBOL SECURITY  
 OLD NEW DELTA SENSIT CODE IDC TICK NAME  
 \*\*\*\*\*

\*\*\* INDUSTRY WEIGHTINGS \*\*\*  
 \*\*\*\*\*  
 CODE INDUSTRY WEIGHT  
 \*\*\*\*\*  
 1 AGRICULTURAL PRODUCTION-CROPS 2.20 %  
 8 FORESTRY 2.81 %  
 12 BITUMINOUS COAL & LIGNITE MIN. 2.88 %  
 13 OIL & GAS EXTRACTION 0.50 %  
 20 FOOD KINDRED PRODUCTS 3.60 %  
 23 APP. & OTH. FIN. PROD. MFAOSH 0.57 %  
 25 FURNITURE AND FIXTURES 1.10 %  
 27 PRINTING PUBLISHING AND A.P. 3.29 %  
 29 PETROLEUM REFIN. & REL. PROD. 3.54 %  
 30 RUBBER AND MISC. PLAST. PROD. 3.02 %  
 32 STONE CLAY GLASS & CONC. PROD. 1.22 %  
 35 MACHINERY EXCEPT ELECTRICAL 5.18 %  
 36 ELE. AND ELE. MACH. 3.05 %  
 37 TRANSPORTATIONS EQUIPMENT 2.33 %  
 38 MEAS. ANAL. & CONT. INST. ETC. 2.63 %  
 45 TRANSPORTATION BY AIR 4.37 %  
 48 COMMUNICATION 16.33 %  
 49 ELECTRIC GAS AND SANIT. SERV. 2.88 %  
 51 WHOLESALE TRADE-NONDR. GOODS 1.83 %  
 52 BUILDING MAT. HARD. GAR. SUPP. 4.63 %  
 60 BANKING 2.63 %  
 61 CRED. AGEN. OTH. THAN BANKS. 3.00 %  
 65 REAL ESTATE 13.35 %  
 67 HOLD. AND OTHER INV. COMP. 3.77 %  
 73 BUSINESS SERVICES 0.99 %  
 75 AUTO. REPAIR SERV. AND GAR. 2.25 %  
 80 HEALTH SERVICES TOTAL 100.00 %

5) BOND RETURN FILE RET\BONDS.PRN  
 6) MONTH SIMULATION BEGINS 8812  
 7) NUMBER OF MONTHS SIMULATED 0  
 8) NUMBER OF STOCKS 56  
 9) NUMBER OF BONDS 0  
 10) STOCKS TO READ PAST 0  
 11) MINIMUM STOCK HOLDING % 100.00  
 12) MAXIMUM INDUSTRY HOLDING % 20.00  
 13) MAXIMUM STOCK HOLDING % 3.00  
 14) NS' AT 24  
 15) NR' TS 24  
 16) BUY/LET 0.00  
 17) TARGET 1.00  
 18) TURN OVER FACTOR 100.00

PORTFOLIO CONSTRUCTED UPON THE FOLLOWING DATA:

MONTH	LIA	PORT	DIF
84	-0.0049200002	-0.0091439467	-0.0042239465
85	0.0073779998	0.0500006266	0.0426262658
86	0.0064139999	0.0290627852	0.0226487853
87	-0.0369299985	-0.0087748254	0.0281551741
88	-0.0717199966	-0.0408916809	0.0308283158
89	-0.0172600001	-0.0077189291	0.0095410710
90	0.0247970000	0.0432149470	0.0184179470
91	-0.0405899994	-0.0083635768	0.0312664226
92	0.0243800003	0.0198711567	0.0442511568
93	0.0573999994	0.0359051302	0.0214948691
94	0.0968960002	0.1454212815	0.2423172891
95	-0.0000500000	-0.0123835430	-0.0123335430
96	0.0181620009	0.0374896154	0.0193276145
97	0.0774770007	0.0865747333	0.0090577326
98	0.0179789998	0.0243982524	0.0064192526
99	0.0464700013	0.0084554564	0.0400145464
100	-0.0297400001	-0.0108487122	0.0188912880
101	0.0134399999	0.0099289045	0.0233689044
102	0.0576109998	0.0509947352	0.0065162646
103	0.0356100015	0.0211411919	0.0144688096
104	0.0013400000	-0.0009915107	0.0003484893
105	0.0458530001	0.0319963023	0.0138566978
106	0.0231340006	0.0141323283	0.0090016723
107	0.0217243396	0.0049838545	0.0267081931

TURNOVER = 0.2540976 %  
 VARIANCE AT OLD WEIGHTS = 14.01026  
 VARIANCE AT NEW WEIGHTS = 14.12339  
 DELTA = -0.8010368 %

\*\*\* BUY \*\*\*

146

WHAT IS CLAIMED IS:

1. A method of correlating a future asset return of a portfolio to future financial liabilities variable over time, comprising the steps of:

(a) determining future payments needed over time to fulfill said future financial liabilities;

(b) selecting a standard index having a standard asset return over time for meeting said future payments; and

(c) analyzing a plurality of assets for identifying selected ones of said assets for providing optimum correlation of the future return of said selected assets to said standard asset return, said optimum correlation achieved by calculating a minimum standard deviation for the difference between the future returns of said plurality of selected assets and said standard asset return, ranking said selected assets from smallest to largest covariance, making an iterative change in weights of said selected assets, reordering the rank of said selected assets and continuing said iterative weight and rank change of said selected assets until the covariance converges to an optimal solution.

2. The method as defined in Claim 1 wherein said future financial liabilities comprises a pension plan.



147

3. The method as defined in Claim 1 wherein said plurality of assets exhibit sensitivity to the same parameter as said future financial liabilities for which said time dependent optimum correlation is determined.

4. A system for correlating a future asset return of a portfolio to future financial liabilities variable over time, comprising:

means for determining future payments needed over time to fulfill said future financial liabilities;

means for selecting a standard having a standard asset return over time to meet said future payments; and

means for analyzing a plurality of assets for identifying selected ones of said assets for providing optimum correlation of the future return of said selected assets to said standard asset return, said optimum correlation achieved by means for calculating a minimum standard deviation for the difference between the future returns of said plurality of assets and said selected standard asset return, said calculating means further ranking said selected assets from smallest to largest covariance, making an iterative change in weights of said selected assets, reordering the rank of said selected assets and continuing said iterative weight and rank change of said selected assets until the covariance converges.

148

5. The system as defined in Claim 4 wherein said future financial liabilities comprise insurance liabilities and pension plan liabilities.

6. The method as defined in Claim 4 wherein a computer performs the functional operations.

1/7

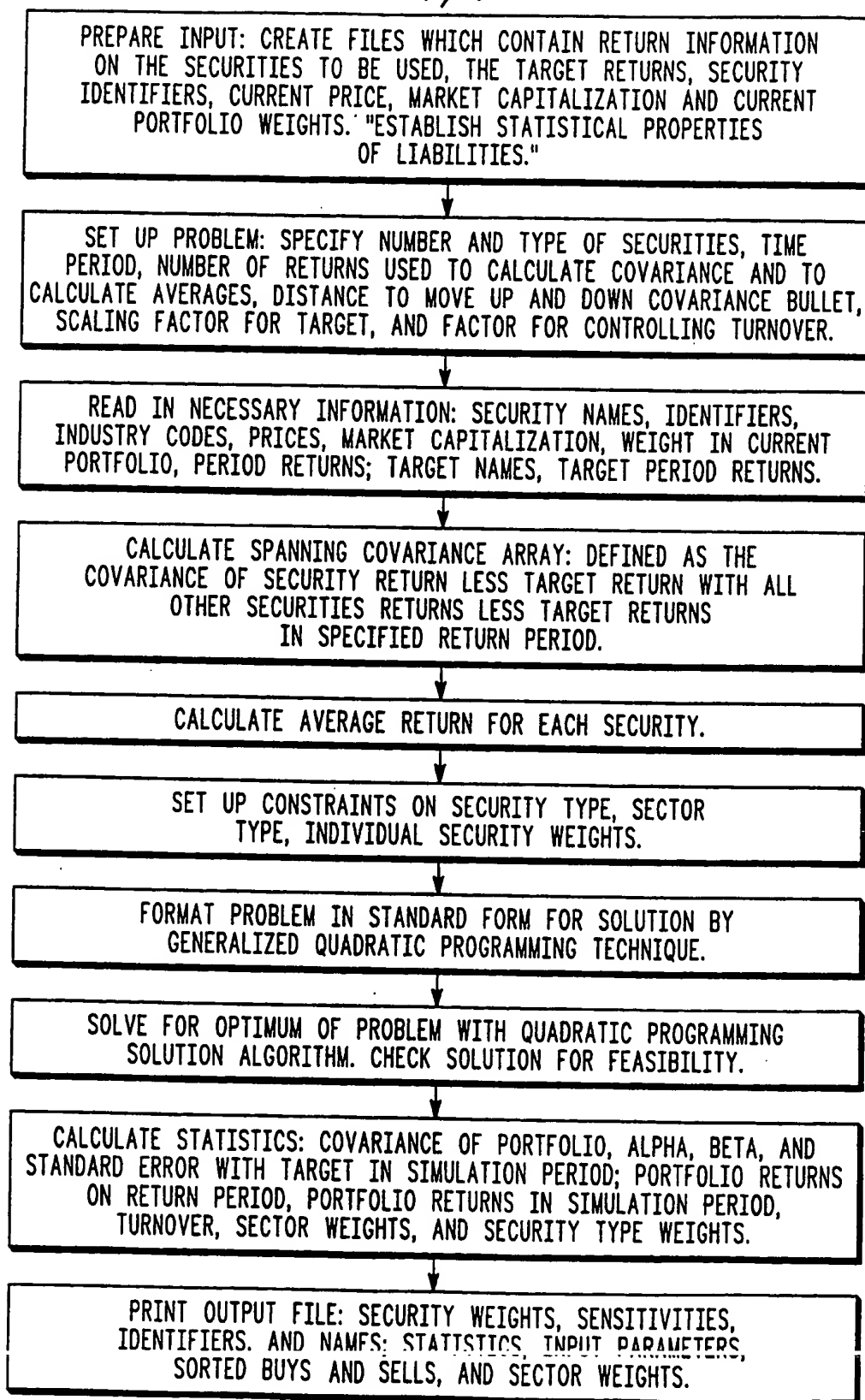
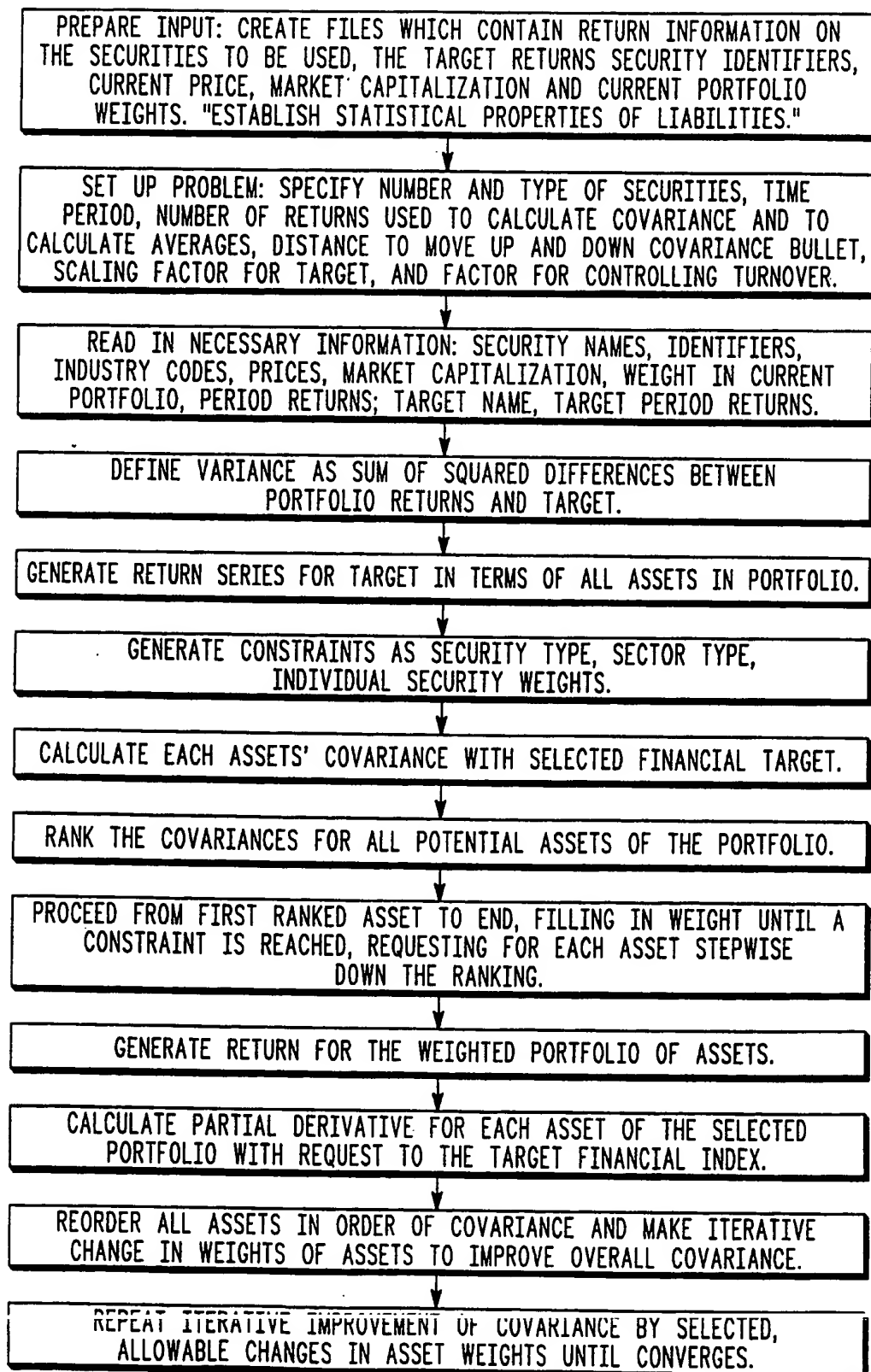


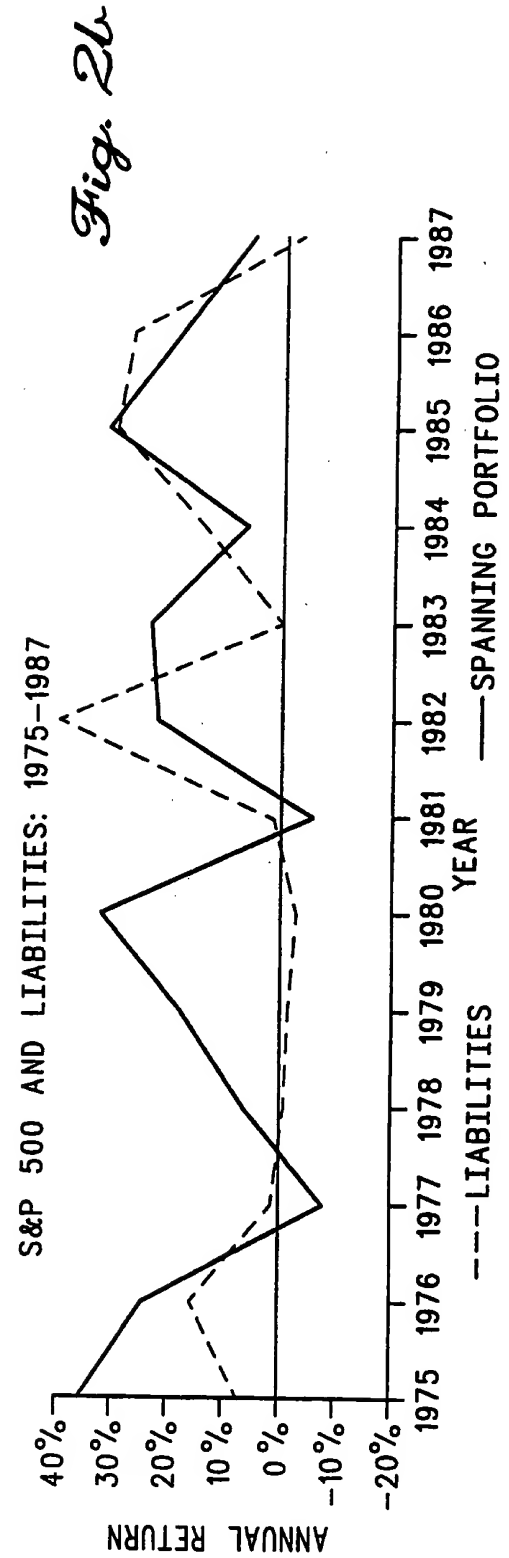
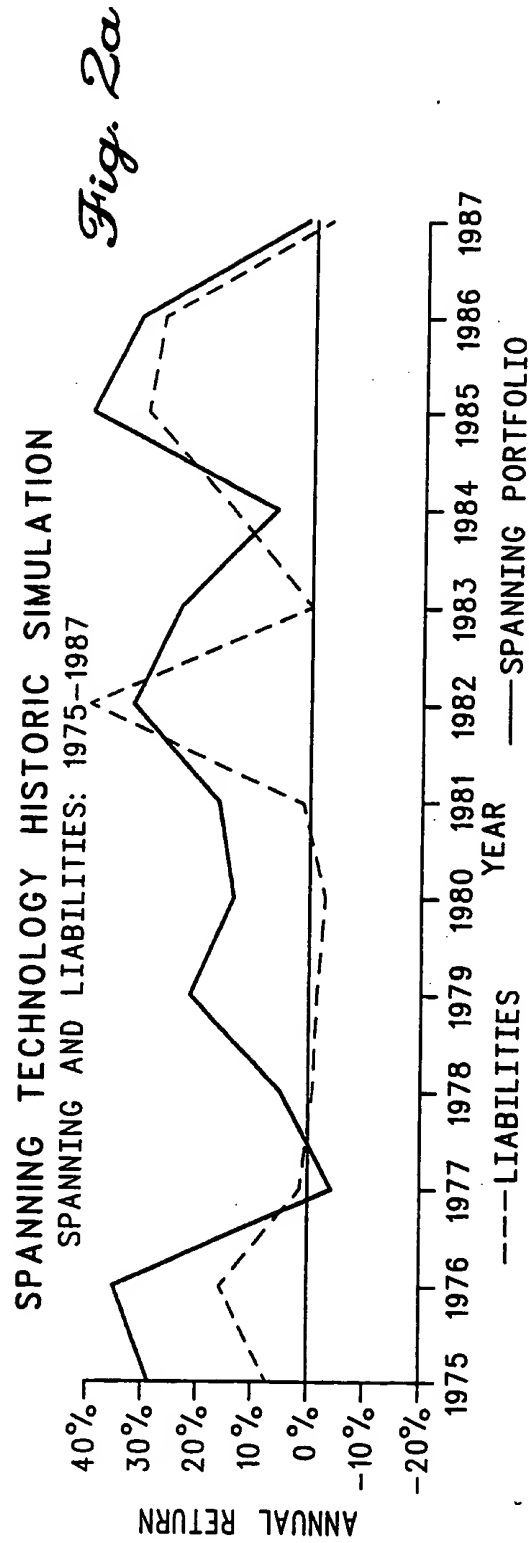
Fig. 1a

2/7

*Fig. 1b*

SUBSTITUTE SHEET

3 / 7

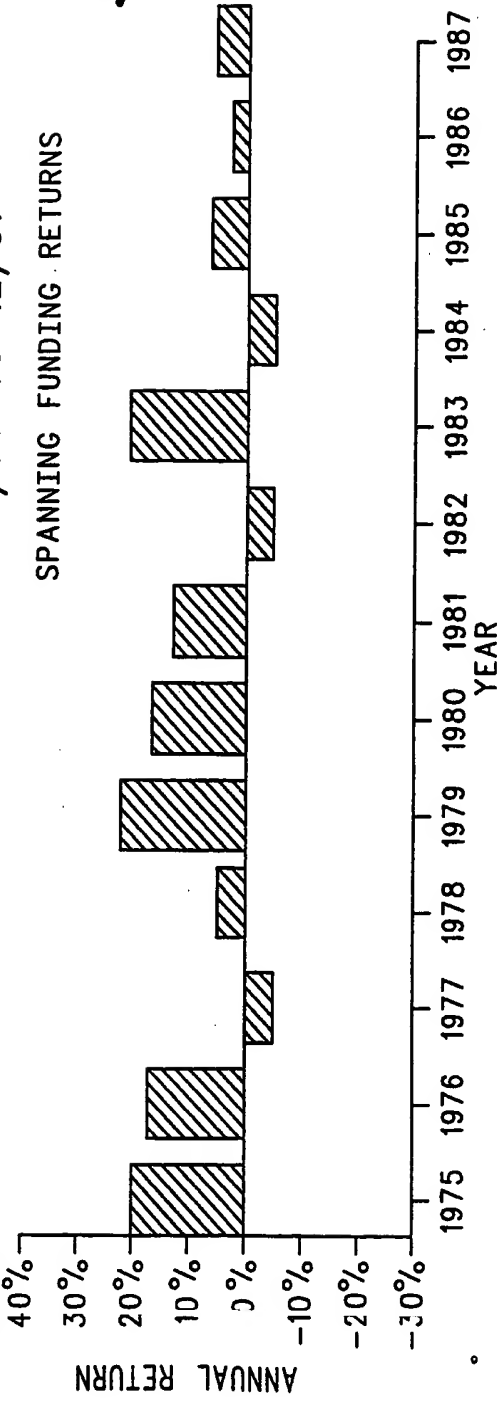


4/7

# SPANNING TECHNOLOGY HISTORIC SIMULATION 1/75 TO 12/87

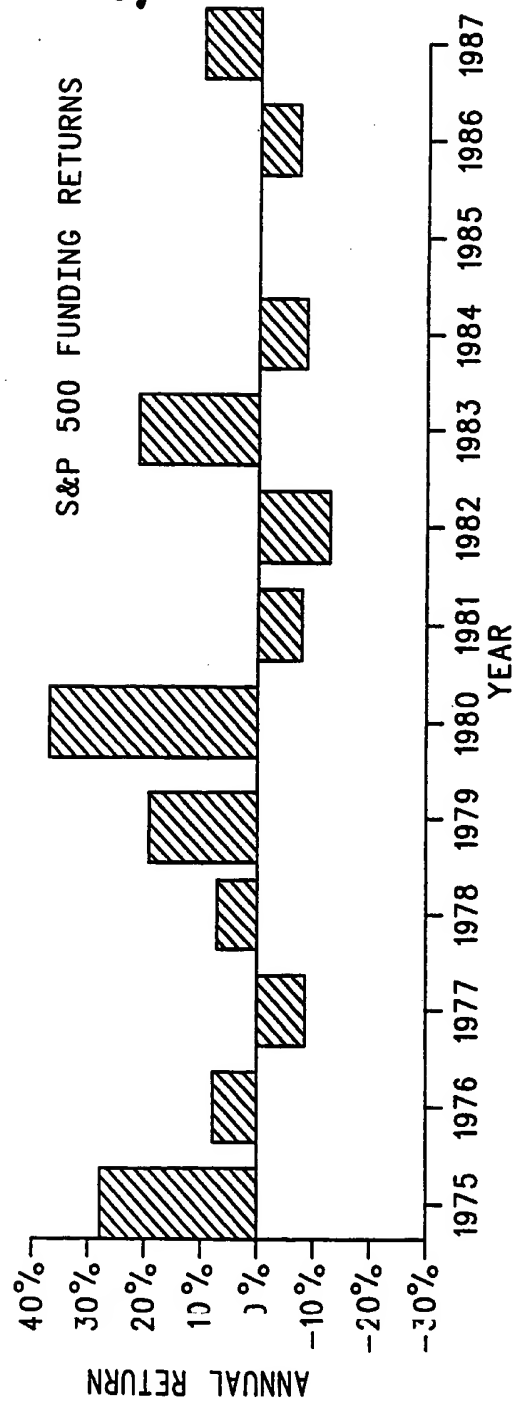
SPANNING FUNDING RETURNS

*Fig. 3a*



S&P 500 FUNDING RETURNS

*Fig. 3b*

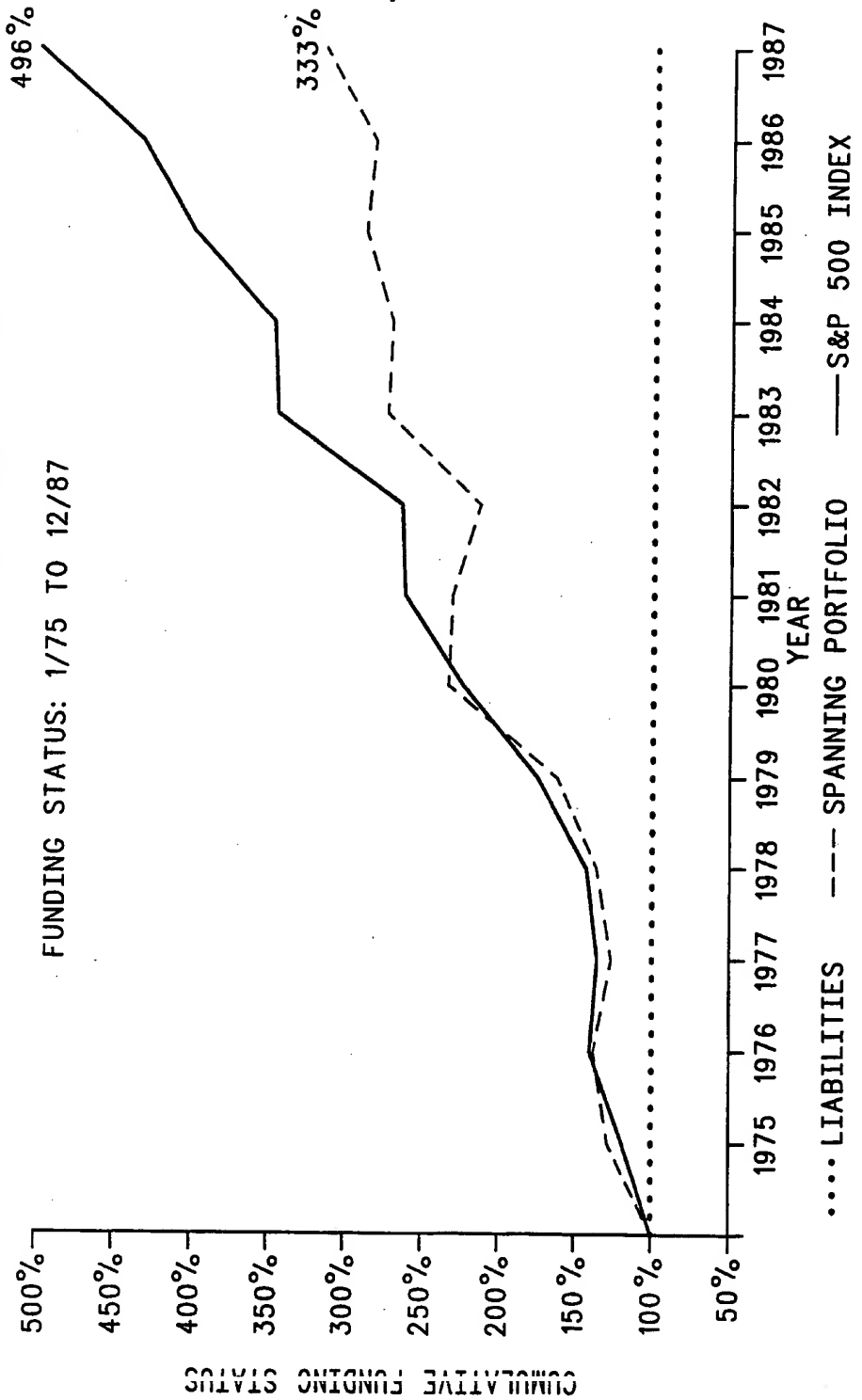


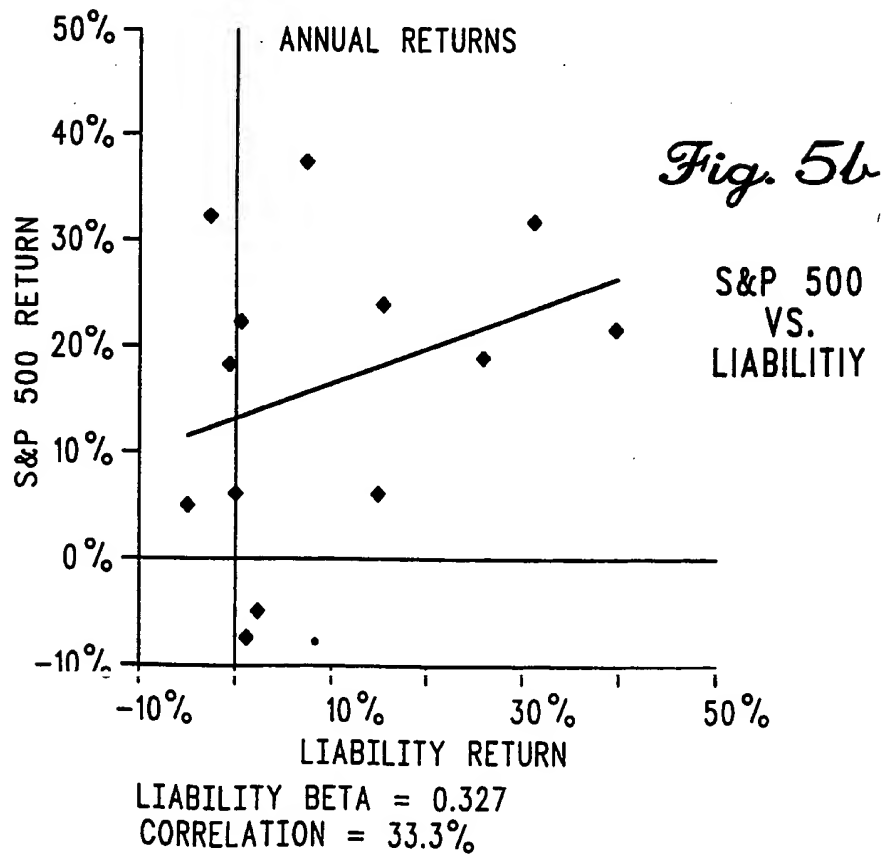
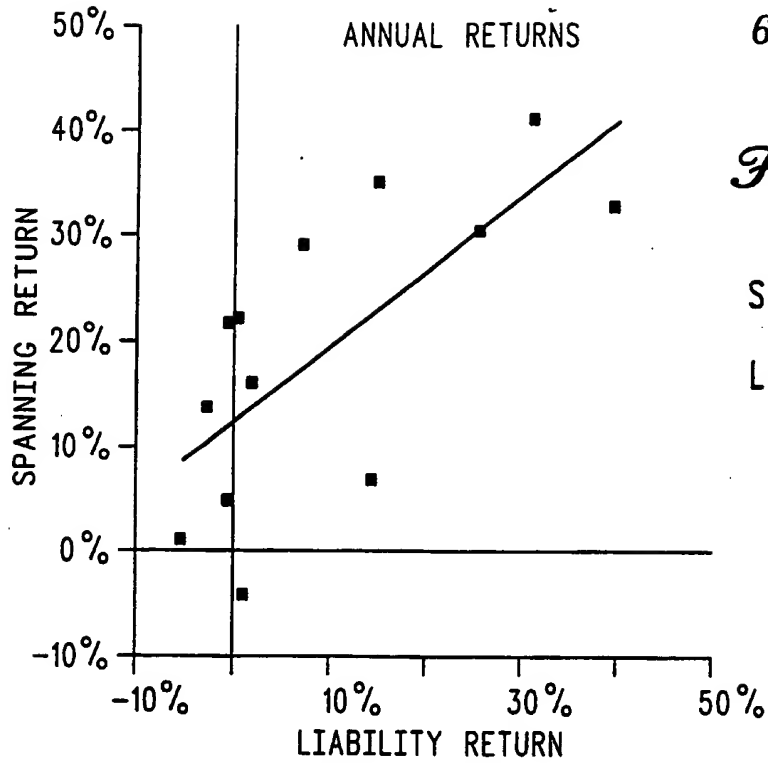
5 / 7

Fig. 4

SPANNING TECHNOLOGY HISTORIC SIMULATION

FUNDING STATUS: 1/75 TO 12/87

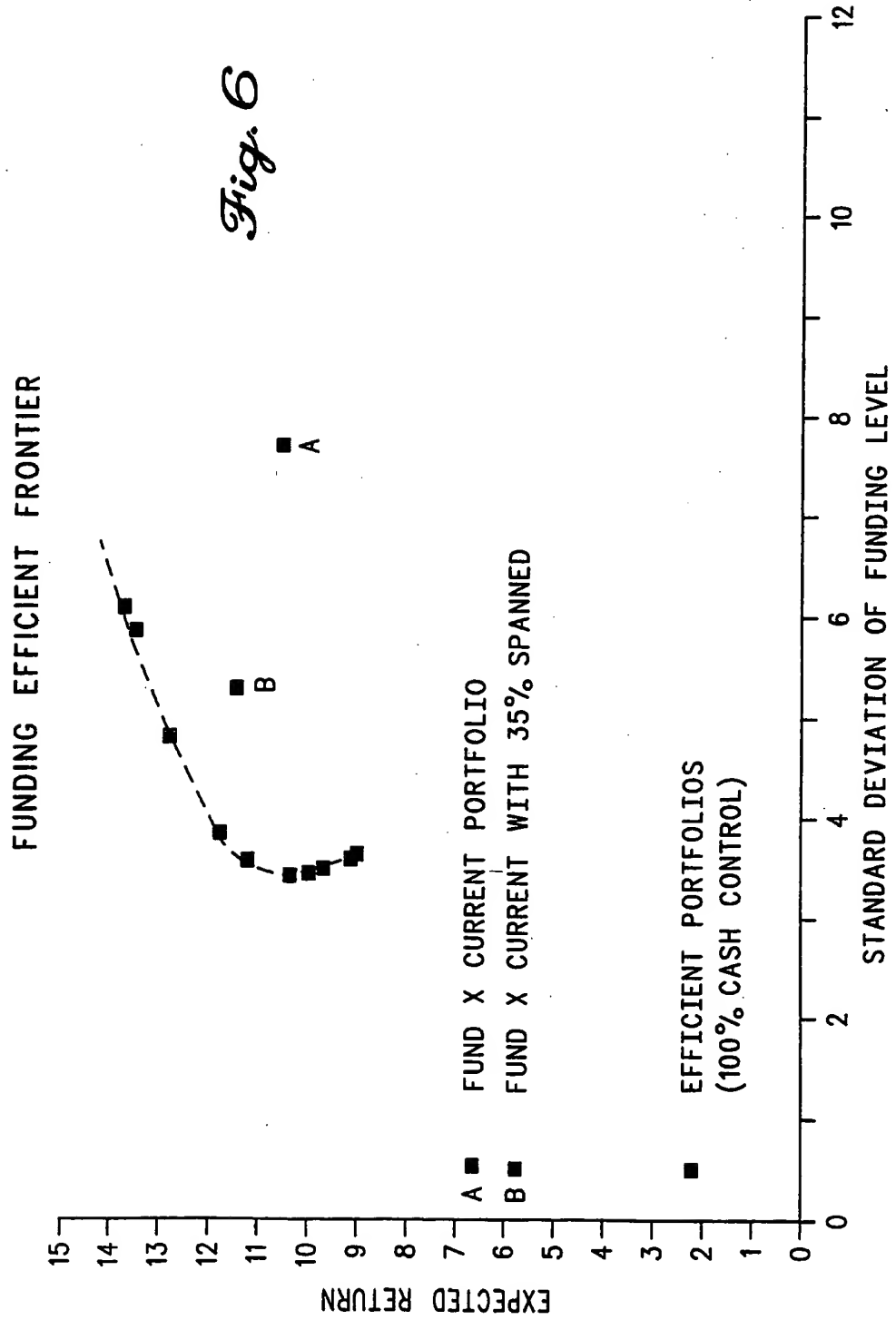






7 / 7

Fig. 6



**SUBSTITUTE SHEET**

# INTERNATIONAL SEARCH REPORT

International Application No PCT/US90/04328

<b>I. CLASSIFICATION OF SUBJECT MATTER</b> (If several classification symbols apply, indicate all) According to International Patent Classification (IPC) or to both National Classification and IPC IPC(5): G06F 15/21 US CL.: 364/401, 408																																
<b>II. FIELDS SEARCHED</b> Minimum Documentation Searched Classification System: US Classification Symbols: 364/401, 408 Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched																																
<b>III. DOCUMENTS CONSIDERED TO BE RELEVANT</b> <table border="1"> <thead> <tr> <th>Category *</th> <th>Citation of Document, <sup>12</sup> with indication, where appropriate, of the relevant passages <sup>13</sup></th> <th>Relevant to Claim No. <sup>14</sup></th> </tr> </thead> <tbody> <tr> <td>A</td> <td>US, A, 4,642,768 (ROBERTS) 10 February 1987</td> <td>1-6</td> </tr> <tr> <td>A</td> <td>US, A, 4,648,038 (ROBERTS et al.) 03 March 1987</td> <td>1-6</td> </tr> <tr> <td>A</td> <td>US, A, 4,722,055 (ROBERTS) 26 January 1988</td> <td>1-6</td> </tr> <tr> <td>A</td> <td>US, A, 4,739,478 (ROBERTS et al.) 19 April 1988</td> <td>1-6</td> </tr> <tr> <td>A</td> <td>US, A, 4,750,121 (HALLEY et al.) 07 June 1988</td> <td>1-6</td> </tr> <tr> <td>A</td> <td>US, A, 4,752,877 (ROBERTS et al.) 21 June 1988</td> <td>1-6</td> </tr> <tr> <td>A</td> <td>US, A, 4,839,804 (ROBERTS et al.) 13 June 1989</td> <td>1-6</td> </tr> <tr> <td>A,P</td> <td>US, A, 4,933,842 (DURBIN et al.) 12 June 1990</td> <td>1-6</td> </tr> <tr> <td>A</td> <td>Haugen, "Modern Investment Theory" published 1985 by Prentice-Hall (New Jersey), Chapters 6-8.</td> <td>1-6</td> </tr> </tbody> </table>			Category *	Citation of Document, <sup>12</sup> with indication, where appropriate, of the relevant passages <sup>13</sup>	Relevant to Claim No. <sup>14</sup>	A	US, A, 4,642,768 (ROBERTS) 10 February 1987	1-6	A	US, A, 4,648,038 (ROBERTS et al.) 03 March 1987	1-6	A	US, A, 4,722,055 (ROBERTS) 26 January 1988	1-6	A	US, A, 4,739,478 (ROBERTS et al.) 19 April 1988	1-6	A	US, A, 4,750,121 (HALLEY et al.) 07 June 1988	1-6	A	US, A, 4,752,877 (ROBERTS et al.) 21 June 1988	1-6	A	US, A, 4,839,804 (ROBERTS et al.) 13 June 1989	1-6	A,P	US, A, 4,933,842 (DURBIN et al.) 12 June 1990	1-6	A	Haugen, "Modern Investment Theory" published 1985 by Prentice-Hall (New Jersey), Chapters 6-8.	1-6
Category *	Citation of Document, <sup>12</sup> with indication, where appropriate, of the relevant passages <sup>13</sup>	Relevant to Claim No. <sup>14</sup>																														
A	US, A, 4,642,768 (ROBERTS) 10 February 1987	1-6																														
A	US, A, 4,648,038 (ROBERTS et al.) 03 March 1987	1-6																														
A	US, A, 4,722,055 (ROBERTS) 26 January 1988	1-6																														
A	US, A, 4,739,478 (ROBERTS et al.) 19 April 1988	1-6																														
A	US, A, 4,750,121 (HALLEY et al.) 07 June 1988	1-6																														
A	US, A, 4,752,877 (ROBERTS et al.) 21 June 1988	1-6																														
A	US, A, 4,839,804 (ROBERTS et al.) 13 June 1989	1-6																														
A,P	US, A, 4,933,842 (DURBIN et al.) 12 June 1990	1-6																														
A	Haugen, "Modern Investment Theory" published 1985 by Prentice-Hall (New Jersey), Chapters 6-8.	1-6																														
<p>* Special categories of cited documents: <sup>15</sup></p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step</p> <p>"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"&amp;" document member of the same patent family</p>																																
<b>IV. CERTIFICATION</b> <table border="1"> <tr> <td>Date of the Actual Completion of the International Search <sup>3</sup></td> <td>Date of Mailing of this International Search Report <sup>3</sup></td> </tr> <tr> <td>20 NOVEMBER 1990</td> <td>25 JAN 1991</td> </tr> <tr> <td>International Searching Authority:</td> <td>Signature of Authorized Officer <sup>16</sup></td> </tr> <tr> <td>ISA/US</td> <td>DAVID HUNTLEY</td> </tr> </table>			Date of the Actual Completion of the International Search <sup>3</sup>	Date of Mailing of this International Search Report <sup>3</sup>	20 NOVEMBER 1990	25 JAN 1991	International Searching Authority:	Signature of Authorized Officer <sup>16</sup>	ISA/US	DAVID HUNTLEY																						
Date of the Actual Completion of the International Search <sup>3</sup>	Date of Mailing of this International Search Report <sup>3</sup>																															
20 NOVEMBER 1990	25 JAN 1991																															
International Searching Authority:	Signature of Authorized Officer <sup>16</sup>																															
ISA/US	DAVID HUNTLEY																															

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**